MODELLING AND SIMULATION 2007

THE EUROPEAN SIMULATION

AND

MODELLING CONFERENCE

2007

ESM® '2007

EDITED BY

Jaroslav Sklenar

Alain Tanguy

Cyrille Bertelle

And

Giancarlo Fortino

OCTOBER 22-24, 2007

ST.JULIAN'S, MALTA

A Publication of EUROSIS-ETI

The European Simulation and Modelling Conference 2007

ST.JULIAN'S, MALTA

OCTOBER 22-24, 2007

Organised by

ETI- The European Technology Institute

Sponsored by

EUROSIS, The European Simulation Society

Co-Sponsored by

DII
Seconda Universita degli Studi di Napoli

Ghent University

ICAR-CNR

SIMULA

and

The University of Malta

Hosted by

Westin Dragonara Hotel

St.Julian's, Malta

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

General Conference Chair

Jaroslav Sklenar, University of Malta, Malta

General Conference Co-Chair

Alain Tanguy, Universite Blaise Pascal, Aubiere, France

Past Conference Chairs

Alexandre Nketsa, LAAS, Toulouse, France
Mario Paludetto, LAAS, Toulouse, France

Journal Publication Chair

Laurence Tianruo Yang, St. Francis Xavier University, Canada.

Local Committee Chairs

John Abela, Department of Computer Science and AI, UOM
Natalie Attard, Department of Statistics & OR, UOM
Monique Inguanez, Department of Statistics & OR, UOM
Adrian Muscat, Department of Communications and Computer Engeneering, UOM
Zdenka Sant, Department of Mechanical Engeneering, UOM
Tony Spiteri Staines, Department of Computer Information Systems, UOM
Fiona Sammut, Department of Statistics & OR, UOM
Charles Sammut, Department of Physics, UOM
David Suda, Department of Statistics & OR, UOM

Track Chair Complex Systems and Self-Organization Modelling

Cyrille Bertelle
LITIS, Le Havre University
Le Havre Cedex, France

Track Chair Petri Nets Simulation

Carmen Veronica Bobeanu
Ghent University
Ghent, Belgium

Track Chair Workshop on Multi-Agent Systems and Simulation (MAS&S)

Giancarlo Fortino
Università della Calabria
Rende, Italy

INTERNATIONAL PROGRAMME COMMITTEE

Methodology and Tools
Pierre l'Ecuyer, Universite de Montreal, Canada
Michel Gourgand, Universite Blaise Pascal, Aubiere, France
Thomas Hanne, Fraunhofer -ITWM, Kaiserslautern, Germany
Celso Hirata, Inst. Tecnologico de Aeronautica, Campos, Brazil
Bjorn Johansson, Chalmers Univ. of Techn, Gotheburg, Sweden
Abder Koukam, Univ.de Technologie de Belfort, Belfort, France
Moreno Marzolla, Universita di Venezia, Mestre, Italy
Carlos E. Pereira, Univ. Fed. do Rio Grande do Sul, Porto Alegre, Brazil
Eric Ramat, LIL, Calais, France
Roberto Revetria, University of Genova, Italy
M. del Puerto Paulo Ruiz, Univesidad de Valladolid, Spain
Bert van Beek, Eindhoven University of Technology, The Netherlands
Abdou Zakari, LITA, Universite de Metz, France

Object-Orientation and Re-Use
Frederic Mallet, University of Nice, Sophia Antipolis, France

Simulation and Artificial Intelligence
Paulo Cortez, University of Minho, Guimareas, Portugal
Charbel Fares, ESIEE, Noisy-le-Grand, France
Adina Magda Florea, Univ. Politehnica of Bucharest, Romania
Vladimir Janousek, Brno University of Technology, Brno, Czech Republic
Esko Juuso, University of Oulu, Finland
Bernd Kleinjohann, University of Paderborn, Germany
Jose Neves, Universidade do Minho, Braga, Portugal
Tudor Niculiu, Univ. Politehnica of Bucharest, Romania
Wolfgang Kreutzer, Univ. of Canterbury, Christchurch, New Zealand
Damien Olivier, Universite du Havre, France
Maja Pantic, Imperial College, London, United Kingdom
Eshan Rajabally, University of Newcastle, Newcastle, United Kingdom
Leon Rothkrantz, TU Delft, The Netherlands
Franciszek Seredynski, Polish Acad.of Science, Warsaw, Poland
Sarawut Suijtjorn, Suranaree Univ.of Technology, Nakhon Ratchasima, Thailand
Jim Torresen, University of Oslo, Norway
Frantisek Zboril, Brno University of Technology, Brno, Czech Republic
Morched Zeghal, Nat. Res.Council Canada, Ottawa, Canada

High Performance Large Scale and Hybrid Computing
Track Chair: Beniamino di Martino, Second University of Naples, Italy
Jan Broeckhove, University of Antwerp, Antwerp, Belgium
Giancarlo Fortino, Universita della Calabria, Rende, Italy
Mohammad Riaz Moghal, Univ.of Engineering and Technology, Mirpur, Pakistan
Jari Porras, Lappeenranta University of Technology, Finland
Simon See, Sun Microsystems Inc, Singapore
Pierre Siron, ONERA, Toulouse, France
Carl Tropper, McGill University, Montreal, Canada
Behrouz Zarei, Sharif University of Technology, Tehran, Iran

INTERNATIONAL PROGRAMME COMMITTEE

Simulation in Education and Graphics Visualization
Magy Seif El-Nasr, Penn State University, Evanston, USA
George Gyftodimos, University of Athens, Greece
Jan Lemeire, VUB, Brussels, Belgium
Qingping Lin, Nanyang Technological University, Singapore
Marco Roccetti, University of Bologna, Italy
Gerardo Silva, University of Santiago, Chile
Guilherme Luis Roehe Vaccaro, PUCRS, Porto Alegre, Brazil
Franz Wotawa, TU Graz, Austria

Simulation in Environment, Ecology, Biology and Medicine
Eduardo Ayesa, CEIT, San Sebastian, Spain
Cyrille Bertelle, LIH, Le Havre, France
Vahid Nassehi, Loughborough University, United Kingdom
Laurent Perochon, INRA, St Genes Champanelle, France
Cezary Orlowski, Technical University Gdansk, Poland
Tarik Ozkul, American University of Sharjah, UAE
Pierrick Tranouez, Labo Info du Havre, France

Analytical and Numerical Modelling Techniques
Ana M. Camacho, UNED, Madrid, Spain
Tom Dhaene, University of Antwerp, Belgium
Clemens Heitzinger, Purdue University, West Lafayette, USA
Eva M. Rubio, UNED, Madrid, Spain
Bruno Tuffin, IRISA, Rennes, France
Sarawut Sujitjorn, Suranaree Univ. of Technology, Nakhon Ratchasima, Thailand

Web Based Simulation
Manuel Alfonseca, Universidad Autonoma de Madrid, Spain
Ammar Al-Khani, VTT Processes, Espoo, Finland
Yan Luo, NIST, Gaithersburg, USA
Jose Machado, University of Minho, Braga, Portugal
Mara Nikolaidou, University of Athens, Greece
Francesco Quaglia, University of Rome I, Italy
Krzysztof Pawlikowski, University of Canterbury, Christchurch, New Zealand
Vaclav Snasel, VSB Technical University of Ostrava, Czech Republic

Agent Based Simulation
Jacinto A. Dávila, Universidad de los Andes, Merida, Venezuela
Eric Gouarderes, UPPA, Pau, France
Frederic Guinand, Universite du Havre, France
Zisheng Huang, Vrije Universiteit Amsterdam, The Netherlands
Jean-Luc Koning, INPG-ESISAR-LEIBNIZ, Grenoble, France
Peter Lawrence, Australian Catholic University, Melbourne, Australia
Ioan Alfred Letia, TU Cluj Napoca, Romania
Paulo Novais, Universidade do Minho, Braga, Portugal
Jan-Torsten Milde, FH Fulda, Germany
Lars Monch, TU Ilmenau, Germany
Isabel Praca, Ist. Superior do Porto, Portugal
Marco Remondino, University of Turin, Italy
Xiaolan Xie, INRIA-LIGPM, Metz, France

INTERNATIONAL PROGRAMME COMMITTEE

Simulation with Petri Nets
Carmen Bobeanu, Ghent University, Belgium
Mauro Iacono, University of Naples II, Italy
Juan de Lara, Univ. Autonoma de Madrid, Spain
Guenter Hommel, TU Berlin, Germany
Stefano Marrone, Seconda Università degli Studi di Napoli, Naples, Italy
Alexandre Nketsa, LAAS-CNRS, Toulouse, France
Mario Paludetto, LAAS-CNRS, Toulouse, France
Ivo Vondrak, Technical University of Ostrava, Czech Republic

Bond Graphs Simulation
Rui Esteves Araujo, DEEC-FEUP, University of Porto, Portugal
Jesus Felez, Univ. Politecnica de Madrid, Spain
Aziz Naamane, DIAM-IUSPIM, Marseille, France
Manuel Rodrigues Quintas, FEUP, University of Porto, Portugal
Andre Tavernier, BioSim, Brussels, Belgium

DEVS
Fabrice Bernardi, University of Corsica, Corte, France
Dirk Brade, FOI, Stockholm, Sweden
Adriano Carvalho, FEUP, University of Porto, Portugal
Alexandre Muzy, Universite de Corse, Corti, France
Fernando Tricas, Universidad de Zaragoza, Spain
Armin Zimmerman, TU Berlin, Germany

Fluid Flow Simulation
Diganta Bhusan Das, University of Oxford, United Kingdom
H.A.Nour Eldin, University of Wuppertal, Germany
Markus Fiedler, Blekinge Institute of Technology, Sweden

SIMULA
Henry Islo, Association of Simula Users
Rob Pooley, Herriot-Watt University, Edinburgh, United Kingdom
Eugene Kindler, University of Ostrava, Ostrava, Czech Republic
Jaroslav Sklenar, University of Malta, Malta
Alain Tanguy, Université Blaise Pascal, Clermont-Ferrand, France

INTERNATIONAL PROGRAMME COMMITTEE

Complex Systems and Self-Organization Modelling
Track Chair: Cyrille Bertelle, LITIS, UFR Sciences et Techniques, Le Havre, France
Program Chairs: Duchamp G.H.E., Paris XIII University, Paris, France
Kadri-Dahmani H., Paris XIII University, Paris, France
IPC:
Aziz-Alaoui M.A., Le Havre University, Le Havre, France
Babkin E., Higher School of Economics (Nihzny Novgorod), Russia
Bennani Y., Paris XIII University, Paris, France
Bertelle C., Le Havre University, Le Havre, France
Michele Colajani, University of Modena, Italy
Cotsaftis M., LTME/ECE, Paris, France
Duchamp G.H.E., Paris XIII University, Paris, France
Guinand F., Le Havre University, Le Havre, France
Hadj Kacem H., Sfax University, Sfax, Tunisie
Iantovics B.L., Petru Maior University of Tg. Mures, Romania
Kadri-Dahmani H., Paris XIII University, Paris, France
Obaidat M.S., Monmouth University, USA
Olivier D., Le Havre University, Le Havre, France
Oqeili S., Al Balqa Applied University, Al Balqa, Jordan
Osmani A., Paris XIII University, Paris, France
Ramos V., Technical University of Lisbon, Lisbon, Portugal
Sadoun B., Al Balqa Applied University, Al Balqa, Jordan
Toumazet F., Paris XIII University, Paris, France
Wang R., Massey University, Albany-Auckland, New Zealand

Workshop on Multi-Agent Systems and Simulation (MAS&S)
Chairs:
Giancarlo Fortino, Università della Calabria, Rende, Italy
Massimo Cossentino, ICAR/CNR and Università di Palermo, Palermo, Italy
Juan Pavon, Universidad Complutense Madrid, Madrid, Spain
IPC
Carol Bernon (France)
Aliaksandr Birukou (Italy)
Paul Davidson (Sweden)
Giuseppe Di Fatta (UK)
Salvatore Gaglio (Italy)
Stephane Galland (France)
Alfredo Garro (Italy)
Paolo Giorgini (Italy)
Marie-Pierre Gleizes (France)
Vincent Hilaire (France)
Franziska Kluegl (Germany)
Abder Koukam (France)
Adolfo Lopez Paredes (Spain)
Vito Morreale (Italy)
Andrea Omicini (Italy)
Wilma Russo (Italy)
Paolo Petta (Austria)
Erwan Tranvouez (France)
Giuseppe Vizzari (Italy)

Simulation in Electrical Systems
Maurice Apap, University of Malta, Malta
Cedric Caruana, University of Malta, Malta
Simon Fabri, University of Malta, Malta
Cyril Spiteri Staines, University of Malta, Malta

EUROPEAN SIMULATION AND MODELLING CONFERENCE 2007

© 2007 EUROSIS-ETI

Responsibility for the accuracy of all statements in each peer-referenced paper rests solely with the author(s). Statements are not necessarily representative of nor endorsed by the European Simulation Society. Permission is granted to photocopy portions of the publication for personal use and for the use of students providing credit is given to the conference and publication. Permission does not extend to other types of reproduction nor to copying for incorporation into commercial advertising nor for any other profit-making purpose. Other publications are encouraged to include 300- to 500-word abstracts or excerpts from any paper contained in this book, provided credits are given to the author and the conference.

All author contact information provided in this Proceedings falls under the European Privacy Law and may not be used in any form, written or electronic, without the written permission of the author and the publisher.

All articles published in this Proceedings have been peer reviewed

EUROSIS-ETI Publications are ISI-Thomson and INSPEC referenced

For permission to publish a complete paper write EUROSIS, c/o Philippe Geril, ETI Executive Director, Ghent University, Faculty of Engineering, Dept. of Industrial Management, Technologiepark 903, Campus Ardoyen, B-9052 Ghent-Zwijnaarde, Belgium.

EUROSIS is a Division of ETI Bvba, The European Technology Institute, Torhoutsesteenweg 162, Box 4, B-8400 Ostend, Belgium

Printed in Belgium by Reproduct NV, Ghent, Belgium
Cover Design by Grafisch Bedrijf Lammaing, Ostend, Belgium

ESM® is a European registered trademark of the **European Technology Institute** under nr: 002433290

EUROSIS-ETI Publication
ISBN: 978-90-77381-36-6

PREFACE

Dear Participant,

It is my pleasure to welcome you to Malta for the 21st annual *European Simulation and Modelling Conference* ESM'2007 organized by EUROSIS and the University of Malta.

As any discussion about the importance of Modelling and Simulation within this community is a waste of time, let me just remind you about two anniversaries that speak for themselves.

40 years ago in May 1967 at the IFIP Working Conference on Simulation Programming Languages that took place in Lysebu hotel near Oslo, Ole-Johan Dahl and Kristen Nygaard from the Norwegian Computing Center presented the paper *Class and Subclass Declaration* describing the main ideas of the then new language SIMULA 67. This was the birth of Object Oriented Programming (OOP) because the language (now called SIMULA) contained all important features of OOP and most of other even much younger OOP languages are just conceptual subsets of SIMULA. Since the very beginning SIMULA was created as both system description and programming language needed to solve optimization and other problems involving complex real world. As a part of this conference there is a workshop and an invited talk on this important anniversary.

So with some exaggeration we may say that Simulation gave the world Object Oriented Programming that today represents the leading programming paradigm. Could there be any more important credit of Simulation and related theories? Yes, there is.

25 years ago Crutzen and Birks published results of their calculations in the paper *The atmosphere after a nuclear war: twilight at noon* (Ambio, 1982) describing the then not recognized effect of dust and smoke released by nuclear explosions. More simulation experiments carried out and publicized by the so-called TTAPS group (Turco, Toon, Ackerman, Pollack and Sagan) confirmed the results; the term *nuclear winter* was born. Similar results were obtained by Aleksandrov and Stenchikov (*On the modeling of the climatic consequences of the nuclear war*, USSR Academy of Sciences, 1983) and others. So Simulation made it clear that even a limited nuclear war cannot have winners. What followed we all know.

For the Simulation community the message is clear. Mathematical and simulation modeling are inseparable, the role of clear interpretation and explanation of simulation results cannot be overemphasized. Let's hope that this conference will contribute to these and many other areas related to Computer Simulation.

Let me express special thanks to the main organizer Philippe Geril and to the keynote and invited speakers. I wish you all a pleasant and fruitful stay in Malta.

Jaroslav Sklenar
ESM2007 General Conference Chair

CONTENTS

Preface ..... **XI**
Scientific Programme **1**
Author Listing **613**

KEYNOTES

Variance Reduction's Greatest Hits
Pierre l'Ecuyer **5**

About the Predictability and Complexity of Complex Systems
Renate Sitte **13**

SIMULA and 40 Years of Object Oriented Programming
Eugene Kindler **20**

METHODOLOGY

SIMULATION METHODOLOGY

Criteria and Methods to Establish the Valid Interaction of a Simulation and its Intended Purpose
Vincent Albert, Alexandre Nketsa, Mario Paludetto and Marc Courvoisier **29**

Application of System Dynamics Methodology for Determination of sustainable Sardine Catch in Eastern Adriatic
Merica Sliskovic, Ante Munitic and Gorana Jelic-Mrcelic **37**

Modelling Discrete Input Probability Distributions
Johann Christoph Strelen **40**

Effective Approximation of Zero-Variance Simulation in a Reliability Setting
Pierre l'Ecuyer and Bruno Tuffin **48**

SIMULATION MODELS

Approaches of an Integrative Simulation Model for Project Development Processes
Sven Tackenberg, Bernhard Kausch, Morten Grandt and Christopher Schlick **57**

Contribution to the Development of Simulation Model of the Ship Steam Boiler
Joško Dvornik and Srđan Dvornik **65**

CONTENTS

Circles Model for Light Metro Rail Analysis
J. Manuel Feliz-Teixeira **68**

An evolution Model of B-spline parametric Surface
Manuel Gonzalez-Hidalgo, Arnau Mir and Gabriel Nicolau-Bestard **74**

Structural Analysis of DAE Models
Adrien Leitold and Miklós Gerzson **81**

MODELLING APPLICATIONS

Reducing the Power Factor Penalty of the Industrial Sector Due to Jordanian Tariff
Nabeel Tawalbeh and Ayman Agha **89**

Modelling of Energy Market Processes in the Distributor Monopolist Structure
Henryk Kaproń and Tomasz Kaproń **92**

Zero Replenishment Phenomenon in a Collaborative Forecasting Planning and Replenishment Environment
Salvatore Cannella and Elena Ciancimino **95**

SIMULATION SOFTWARE TOOLS

SIMULATION TOOLS

A Simulation Study of an automated System for a Shoe Factory using Arena
António Carvalho Brito and Paulo Sá Marques **107**

VisualRTJSim – A Simulation Tool for Real-Time Java
Werner Pohlmann and Markus Rabenberger **112**

SIMULA

SIMULA Tool for Simulation-Based Optimization
Jaroslav Sklenar **119**

New Flexible Simulation Tool in SIMULA
Ivan Krivy and Eugene Kindler **124**

SIMULA Features, Extensions and Simulation Tools
Alain Tanguy **129**

SCICOS/MODELICA

Thermalhydraulics applications developed in Scicos
Masoud Najafi and Daniel Bouskela **137**

Modeling complex systems with Modelica in Scicos: Application to mean value spark engine
Masoud Najafi and Zakia Benjelloun-Dabaghi **142**

Implementation in Modelica of a Virtual Lab for testing Washing Machine Designs
Carla Martín-Villalba, Alfonso Urquia, Sebastián Dormido and Felix Martinez **147**

PETRI NETS SIMULATION

Algebraic Techniques & Symmetries: an Efficient Simulation Approach for Well-Formed Petri Nets
Lorenzo Capra **155**

Evaluation of Vehicle Routing Solutions by Time Petri Nets under Condition of Travel Time Uncertainty
Gerrit K. Janssens, An Caris, Katrien Ramaekers and Dencho N. Batanov **163**

Simulation and Analysis of Condition/Event Petri Nets Using Software Tool CESim
Petr Novosad and Milan Ceska **170**

Performance Analysis of a Photonic Manhattan Street Network using Colored Petri Nets
Ajay Anand, B V R. Reddy and Navin Rajpal **172**

A High-level Petri Net Based Modeling Approach for Risk Management in Supply Chain Networks
Gonca Tuncel and Gülgün Alpan **178**

ANALYTICAL AND NUMERICAL MODELLING TECHNIQUES

PERFORMANCE OPTIMIZATION

Two Monte Carlo Studies for Latent Class Segmentation Models
Liberato Camilleri **191**

CONTENTS

Simulation for Optimizing the Generation of Electric Energy Considering Renewable Sources
Eugeniusz M. Sroczan **199**

PROJECT MANAGEMENT SIMULATION

Project Management Processes modelling Using Petri Nets
Sarka Kvetonova and Jitka Kreslikova **205**

Strategic Manpower Planning for the Belgian Defence: A Markovian Steady-State Model
Filip Van Utterbeeck, Raf Van Loock and Dirk Van Oudheusden **210**

A Batch Server with Service Times Dependent on the Number of Served Customers
Dieter Claeys, Joris Walraevens, Koenraad Laevens and Herwig Bruneel **214**

Two-Band k.p. Model for the Conduction Band in Silicon
Viktor Sverdlov, Gerhard Karlowatz, Hans Kosina and Siegfried Selberherr **220**

PROCESS SIMULATION

Modelling Hardwood Delignification in continuous Downflow Lo-Solids™ Cooking
Minna Palosaari and Timo Ahvenlampi **227**

Data Analysis and Simulation for Queueing Systems
Luigi Rarità and Eliza Trapel **230**

Numerical Simulation of Stereolithographic Processes
João M. Matias and Paulo J. Bártolo **235**

COMPLEX SYSTEMS SIMULATION

CONCEPTUAL MODELS AND TOOLS FOR COMPLEXITY ANALYSIS

A simulation-optimisation framework for determining a best strategy in inventory decision making and demand forecasting for intermittent demand
Katrien Ramaekers and Gerrit K. Janssens **249**

Invariant Algebraic Manifolds of Differential Systems
Jean-Marc Ginoux and Bruno Rossetto **255**

On the Decentralized Dynamic Graph Coloring Problem
Antoine Dutot, Frederic Guinand, Damien Olivier and Yoann Pigné **259**

CONTENTS

Control on System Diffusion Using Genetic Automata
Luai Jaff, Cyrille Bertelle and Gérard H.E. Duchamp **262**

Building upon Fast Multipole Methods to detect and model organizations
Pierrick Tranouez and Antoine Dutot **267**

Design and Validation of a Distributed Computation Environment for Mobile Devices
Ian Oliver **272**

TRANSPORT AND TRAFFIC FLOW

A Simulation Approach to the Analysis of Intermodal Freight Transport Networks
An Caris, Gerrit K. Janssens and Cathy Macharis **283**

A Non-Linear transportation Problem with One Hundred Variables
William Conley **287**

Investigation of the Spatial Distribution Algorithm of the Traffic Flow Analysis and of the Entity Flow Phase analysis
Gábor Lencse and László Muka **291**

An Integrated Environment for Urban Traffic Flow Simulation Using Cellular Automata
Ourania Hatzi, Stephanos Thomas, Vassilis Dalakas, Mara Nikolaidou and Dimosthenis Anagnostopoulos **296**

ADAPTATION AND SELF-ORGANIZATION APPLICATIONS

Time-Optimal Adaptive Target tracking by a Mobile Robot
Krzysztof Skrzypczyk **303**

On the Use of Generalized Derangements for Schelling's Model of Segregation
Rawan Ghnemat, Gerard H.E. Duchamp and Cyrille Bertelle **306**

Emotion: Appraisal-coping Model for the "Cascades" Problem
Karim Mahboub, Evelyne Clément, Cyrille Bertelle and Véronique Jay **309**

On the Use of Ant Colony in Associative Classification
Kifaya Qaddoum, Fadi Thabtah and Cyrille Bertelle **315**

CONTENTS

ARTIFICIAL INTELLIGENCE, NEURAL NETS AND FUZZY SYSTEMS

ARTIFICIAL INTELLIGENCE

Linear and Integer Programming Large Scale Heuristic for Strips Planning
Adam Galuszka.....**321**

Context-based Fuzzy Clustered-Oriented Decision Trees
Zenon A. Sosnowski **326**

NEURAL NETS

Force-based Visualization for Competitive Learning methods
Dušan Fedorèák and Ivo Vondrák **331**

Evaluation of Forest Tree Distribution Model using Artificial Neural Networks
Salvis Dagis **336**

AGENT BASED SIMULATION

AGENT BASED PLANNING

Agent Planning and Scheduling in Distributed Manufacturing Systems
Kamel Benaissa and Daniel Diep.....**345**

Global Multi-Plant Flexible Networks Simulator
Eduardo Castellano, Iñaki Zugasti, Jone Uribetxeberria and
Juan Manuel Besga **348**

Modeling Strategic and Operational Decision Making an Agent Based Model of Electricity Producers
Emile Chappin, Gerard Dijkema, Koen van Dam and Zofia Lukszo.....**356**

VILLON-Agent Based generic Simulation Model of Transportation Logistic Terminals
Norbert Adamko, Valent Klima and Antonin Kavička **364**

INTELLIGENT AGENT MODELLING

Towards a psychological perception in agent modelling
Jean-Michel Auberlet **371**

CONTENTS

A Model of prospective Memory and Habit Phenomena calibrated with dynamic Field Data
Robert Tobias, Christian Würzebesser and Hans-Joachim Mosler.....**373**

Organisational Learning via Interactive Process Simulation in AGE
N. B. Szirbik and G.B.Roest.....**381**

SOCIAL AGENTS

MACSEM Improvement for Agent Based Coalitions
Isabel Praca, Andre Andrade, Hugo Morais, Carlos Ramos and Zita Vale.....**389**

Modelling Group Decision Simulation through Argumentation
Goreti Marreiros, Paulo Novais, José Machado, Carlos Ramos and José Neves.....**394**

Technical Diffusion by Social Means: an Agent Based Framework for District Heating
Marco Remondino.....**402**

AGENT BASED MODELS

Agent-Based Model of Methicillin-Resistant *Staphylococcus aureus* and Antibiotics in Batch Culture
James T. Murphy, Ray Walshe and Marc Devocelle **409**

Adaptative Dichotomic Optimization: a New Method for the Calibration of Agent-Based Models
Benoît Calvez and Guillaume Hutzler **415**

MULTI AGENT SYSTEMS AND SIMULATION

AGENT ORIENTED METHODOLOGIES

Multi-Agent Systems Formalization using Formal Method DPLA
Henrikas Pranevicius, Dalius Makackas and Agne Paulauskaite-Taraseviciene.....**427**

Living Design: Simulation for Self-Designing Agents
Sylvain Lemouzy, Carole Bernon and Marie-Pierre Gleizes **432**

Agent Based Modeling and Simulation of Cooperative Content Distribution Networks
Giancarlo Fortino, Alfredo Garro, Samuele Mascillaro and Wilma Russo.....**437**

CONTENTS

MULTI AGENT SYSTEMS AND MODELS

A Multiagent View of Grid Computing for biological Simulations in Virtual Reality
Mikael Bourhis and Vincent Rodin **447**

Multi-Agent Systems and Negotiation in a Group Forming Scenario
Ingo Stengel, Udo Bleimann, Mathias Kreider, Martin Kanold and Jeanne Stynes **452**

Multi-Agent Simulation of an Educational Collaborative Web System
Álvaro Barbero, Mario Salvador González-Rodríguez, Juan de Lara and Manuel Alfonseca **458**

SIMULATION ARCHITECTURES AND APPLICATIONS

Norm Selection through Simulation in a Resource-Gathering Society
Daniel Villatoro and Jordi Sabater-Mir **467**

A Simulation based Agent Architecture for supporting Decision Making in Dynamic Environments
Fabio Palopoli and Sergio Greco **475**

WEB BASED SIMULATION

A Performance Test Platform
R. Heinzl, G. Mach, P. Schwaha and S. Selberherr **483**

Labtool - A Managing Software for Computer Courses
R. Heinzl, G. Mach, P. Schwaha and S. Selberherr **488**

Electro-Biological Simulation using a Web Front-End
P. Schwaha, R. Heinzl, G. Mach, C. Pogoreutz, S. Fister and S. Selberherr **493**

SIMULATION IN EDUCATION

Interactive Refinement of a Material Flow Simulation Model by Comparing Multiple Simulation Runs in One 3D Environment
Matthias Fischer, Christoph Laroque, Daniel Huber, Jens Krokowski, Bengt Mueck, Michael Kortenjan, Mark Aufenanger and Wilhelm Dangelmaier **499**

A Multi-Mode Mesh Generation Approach for Scientific Computing
F. Stimpfl, R. Heinzl, P. Schwaha and S.Selberherr **506**

Modular L-Systems: Generating Procedural Models using an Integrated Approach
J. L. Hidalgo, E. Camahort, F. Abad and R. Vivo **514**

A Market Model for the Didactic Industrial Management Simulation
Peter Lawrence **519**

A Simulation for Experiential Learning in Supply Chain Management
Peter Lawrence and Ann Lawrence **525**

BIOLOGICAL SIMULATION

Life Cycles and Competition in Modelling of Artificial and Biological Control System
Ivars Mozga, Uldis Grunde-Zeiferts, Sandis Sudars and Egils Stalidzans **533**

Mitochondrial OxydoReduction Simulation using Multi-Agent System
N. Parisey, J.P. Mazat and M. Beurton-Aimar **537**

Verification of Potential Binding Sites obtained with Stochastic Roadmap Simulation
Marcin Pacholczyk and Zbigniew Starosolski **542**

Forest Fire Spread Modelling: A Practical GIS-based Approach
Yves Dumond **547**

FLUID FLOW SIMULATION

CANDU 6 Nuclear Power Plant Containment Modeling
Ilie Prisecaru, Daniel Dupleac and Mihaela Biro **555**

An evaluation of the influences of compressibility, body forces, rheology and wall slip on the computer modelling of injection moulding of thermoplastics
Vahid Nassehi, Abhijeet Kulkarni and Ramin Salemi **560**

HIGH PERFORMANCE SIMULATION

Simulation of the Emmil E-Marketplace Model in See-Grid using the P-Grade Portal
Peter Kacsuk, Gábor Hermann, Ákos Balasko and
Lívia Kacsukné Bruckner **569**

CONTENTS

Decision Support Method for Efficient Sequential and Parallel Simulation: Time Decomposition in Modified Conceptual Models
László Muka and Gábor Lencse **574**

Case Study: Performance Evaluation for Communication Strategies in Distributed Embedded Systems
Falk Berger, Bernd Däne, Alexander Pacholik and Wolfgang Fengler..... **582**

Optimizing the Time Warp Protocol with Learning Automata
Jun Wang and Carl Tropper..... **586**

ELECTRICAL SIMULATION

Projection Methods for Integral Equations in the Modelling in the Electrical Engineering
Lechoslaw Hacia and Adam Marlewski..... **595**

Kahn Based Performance Model within Co-Design Flow
Kamel Smiri, Mohamed Moalla, Hager Harbegue, Abderrazek Jemai and Ahmed Chiheb Ammari..... **600**

Component Hybrid Dynamic Nets General Representation using UML
S.Krabatou, J.Saadi and A.Zakari..... **603**

Simulation of integrated control system for climate comfort
Mariusz Nowak and Andrzej Urbaniak..... **606**

SCIENTIFIC PROGRAMME

KEYNOTES

VARIANCE REDUCTION'S GREATEST HITS

Pierre L'Ecuyer

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal (Québec), H3C 3J7, CANADA

ABSTRACT

Monte Carlo simulation is an incredibly versatile tool for studying complex stochastic systems. By replicating the simulation several times independently, one can in principle estimate performances measures of the system to arbitrary accuracy. Decisions and operating rules can also be optimized via simulation. A major drawback, however, is that the method converges very slowly and often requires an excessive amount of computing time.

Efficiency improvement methods provide ways of either reducing the required computing time for a given target accuracy, or of obtaining an estimator with better accuracy for a given computational budget. Variance reduction is the primary way of improving efficiency. Key ideas for variance reduction were already introduced in the early days of the Monte Carlo method, in the late forties, at Los Alamos. Since then, enormous progress has been made in our understanding of these methods.

This paper is a guided tour of five different methods that can make a huge difference in the accuracy of simulation estimators. They can reduce the variance (or improve the efficiency) by an arbitrary large factor. In some situations, this type of efficiency improvement is essential for the simulation approach to be viable. We discuss *common random numbers* and their synchronization for comparing similar systems, for derivative estimation, and for optimization, *importance sampling* for rare-event simulation, exploiting auxiliary information via *control variates*, smoothing estimators via *conditional Monte Carlo*, and reducing the noise via generalized *antithetic variates or quasi-Monte Carlo*. We give examples where a clever use of these methods can make a huge difference in the required computing time for a given target accuracy.

INTRODUCTION

Suppose we want to estimate $\mu = \mathbb{E}[X]$ for some random variable X having a very complicated distribution that we can only simulate. With the Monte Carlo method, we generate n independent realizations of X, take their average $\bar{X}_n$ and sample variance S_n^2, and use this information to compute a confidence interval on μ. For a given n, the accuracy of the method depends on $\sigma^2 = \mathrm{Var}[X]$. In this paper, we focus on ways of constructing an alternative estimator having the same expectation as X, but with much smaller variance. Other factors can also be taken into account to measure the quality of an estimator; for example the bias (when it differs from zero) and the computational cost of the estimator (e.g., in terms of CPU time) (Glynn and Whitt 1992). Here, we assume that there is no bias, we disregard the computational cost (or assume that it does change much when we change the estimator), and we focus on the variance. The *efficiency* of an estimator is often defined as the inverse of the product of the mean square error by the expected computing cost. With our simplifications, if becomes equivalent to one over the variance.

A selection of five variance reduction methods are discussed here: common random numbers (CRN), importance sampling (IS), control variates (CV), conditional Monte Carlo (CMC), and randomized quasi-Monte Carlo (RQMC). For each of these methods, the variance can be reduced by an arbitrary large factor, and one can construct examples where the variance can be reduced to zero. Each of them has also been shown to provide a very large efficiency improvement (by a factor in the thousands or millions or more) in at least one realistic application. There are also situations (e.g., in rare-event simulation) where no meaningful estimator could be obtained without these techniques. Additional numerical illustrations will be given in the talk.

There are of course other useful variance reduction (or efficiency improvement) methods than these five. To probe further, we refer the reader to Fishman (1996), Glasserman (2004), Glynn (1994), L'Ecuyer (1994), and L'Ecuyer (2008).

COMMON RANDOM NUMBERS

Simulation is used frequently to compare the performance of similar systems, often for the purpose of optimization. In that context, we often want to estimate differences of the form $\mu_2 - \mu_1$ by $\Delta = X_2 - X_1$, where $\mu_k = \mathbb{E}[X_k]$ for $k = 1, 2$. This estimator has variance

$$\mathrm{Var}[\Delta] = \mathrm{Var}[X_1] + \mathrm{Var}[X_2] - 2\mathrm{Cov}[X_1, X_2],$$

which is minimized by maximizing the covariance between X_1 and X_2. If X_k has (fixed) distribution function F_k for $k = 1, 2$, then a classical result of Fréchet (1951) tells us that the covariance is maximized by taking $X_k = F_k^{-1}(U)$ where $U \sim U(0,1)$ is a *common* random variable, uniformly distributed over the interval $(0,1)$. For complex simulations, generating X_k by inversion from a single uniform like this is typically impractical, because F_k is much too complicated. What can be done, however, is to use exactly the same streams of uniforms to drive the simulation for the two systems, try to use them for the same purpose in both systems (this is called *synchronization*), and generate all random variates by inversion. This technique is known as *common random numbers* (CRN). An intuitive interpretation is that by sharing the same "random noise", the observed differences in performance will be due mainly to real differences between the two systems, and not to the different choice of random numbers. If for each uniform U_j used in the simulation, both X_1 and X_2 are monotone function of U_j in the same direction (increasing or decreasing), and they are not independent, then one can prove that $\mathrm{Cov}[X_1, X_2] > 0$. The variance is then guaranteed to be reduced. There are situations where we want to compare not only two, but hundreds or even thousands or millions of similar systems. These systems usually differ by the values taken by some decision variables.

An interesting setting is when the performance of interest is a mathematical expectation which is a smooth function of some parameter θ, say $\mu(\theta)$, and we want to estimate the derivative $\mu'(\theta) = \partial\mu(\theta)/\partial\theta$ (or the gradient, if θ is a vector). This occurs frequently for sensitivity analysis with respect to θ if we have doubts about the exact value of θ, or within an optimization procedure in which θ is a continuous decision variable (L'Ecuyer 1991, L'Ecuyer and Perron 1994, L'Ecuyer and Yin 1998, Fu 2006)). Gradient estimates are also required for the implementation of hedging strategies of a financial portfolio by trading the underlying assets (Glasserman 2004; Chapter 7).

Suppose $\mu(\theta) = \mathbb{E}[X(\theta, \mathbf{U})]$ for some random variable $X(\theta) = X(\theta, \mathbf{U})$ that depends on θ and on a vector $\mathbf{U}$ of independent uniform random variables, and which can be generated by simulation. We estimate $\mu'(\theta)$ by the finite difference

$$D(\delta) = \frac{X(\theta + \delta, \mathbf{U}_2) - X(\theta, \mathbf{U}_1)}{\delta}$$

for some small $\delta > 0$, where $\mathbf{U}_1$ and $\mathbf{U}_2$ are two vectors of uniforms.

Proposition 1 *The following is proved is L'Ecuyer and Perron (1994) and L'Ecuyer (2008):*

(i) If $\mathbf{U}_1$ *and* $\mathbf{U}_2$ *are independent, then*

$$\delta^2 \mathrm{Var}[D(\delta)] \to 2\mathrm{Var}[X(\theta)]$$

when $\delta \to 0$. *This means that the variance of* $D(\theta)$ *increases to infinity at rate* $1/\delta^2$.

(ii) If $\mathbf{U}_1 = \mathbf{U}_2 = \mathbf{U}$ *(CRNs), if* $X(\theta, \mathbf{U})$ *is continuous in* θ *and differentiable almost everywhere, and if* $D(\delta)$ *is uniformly integrable (uniformly in* θ*), then* $\mathrm{Var}[D(\delta)]$ *remains bounded when* $\delta \to 0$.

(iii) Suppose now that $\mathbf{U}_1 = \mathbf{U}_2 = \mathbf{U}$ *but that* $X(\theta, \mathbf{U})$ *is* not *continuous in* θ. *If the probability that* $X(\cdot, \mathbf{U})$ *is discontinuous in the interval* $(\theta, \theta + \delta)$ *converges to 0 as* $O(\delta^\beta)$ *when* $\delta \to 0$, *and if* $X^{2+\varepsilon}(\theta)$ *is uniformly integrable for some* $\varepsilon > 0$, *then* $\mathrm{Var}[D(\delta)] = O(\delta^{\beta-2-\varepsilon})$, *for any* $\varepsilon > 0$, *when* $\delta \to 0$.

Note that the ratio of variances (and the efficiency improvement factor) between any two of these cases can become arbitrary large when $\delta \to 0$. For example, if the bound in (ii) and $\mathrm{Var}[X(\theta)]$ are both equal to 1, and if we take $\delta = 10^{-4}$, then $\mathrm{Var}[D(\delta)]$ is 2×10^8 (200 millions) times larger with (i) than with (ii). This means that we would need 200 millions times more simulation runs to reach the same accuracy with independent random numbers (case i) than with CRNs (case ii).

When case (ii) holds, we can sometimes take the stochastic derivative $X'(\theta) = \lim_{\delta \to 0} D(\theta)$ as an (unbiased) estimator of $\mu'(\theta)$ (L'Ecuyer 1990, Glasserman 1991; 2004). This is the best possible situation, provided that this $X'(\theta)$ is not too difficult to compute. Glasserman (1991, 2004) provides several examples. Sometimes, we may intentionally change the definition of $X(\theta)$ to make it continuous and benefit from (ii), e.g., by replacing some of its components by conditional expectations (see the section on CMC). For example, if $X(\theta)$ counts the customer abandonments in a queueing system, we may replace each indicator of abandonment (which takes the value 0 or 1) by its conditional expectation (the probability that this customer abandons) given its waiting time.

Case (iii) shows that using common random numbers can still provide very substantial benefit even we cannot make $X(\theta)$

continuous. L'Ecuyer (2008) studies further refinements of case (iii) and gives examples. L'Ecuyer and Buist (2006) study a model of a large telephone call center for which $X(\theta)$ is the number of incoming calls who had to wait less than 20 seconds on a given day, and θ is a parameter (e.g., the mean) of the service time distribution. In that case, $X(\theta)$ can only take integer values, so it cannot be continuous unless it is a constant, in which case $D(\theta) = 0$. L'Ecuyer and Buist (2006) prove that with well-synchonized common random numbers, (iii) applies with $\beta = 1$. In their empirical results with $\delta = 10^{-3}$, the variance is reduced by a factor of about 30,000 compared with independent random numbers.

Their example also illustrates the importance and difficulties of synchronization. There are four types of random numbers in their model; they are used to generate the following random variables: (a) the busyness factor of the day, in the morning (this factor determines the time-varying arrival rate for the day); (b) the interarrival times between successive calls; (c) the call durations, or service times; and (d) the patience times (as soon as the waiting time of a call reaches its patience time, the call abandons). It is important to underline that when simulating the model for two different values of θ, the random numbers are not always used in the same order. For example, a given service time may be generated before a given arrival in one case, and after that arrival in the other case. Moreover, some customers may abandon in one case (so there is no need to generate their service time) and not in the other case. And a call may start its service immediately upon arrival in one case (so there is no need to generate its patience time) and not in the other case. To maintain the synchonization, we use four different streams of random numbers, one stream for each type. Modern simulation tools like Arena, Simul8, Witness, or SSJ (for example) provide such multiple streams of random numbers, with multiple substreams for each stream, and tools to rewind back to the start of the current stream or substream, or to jump ahead to the next substream (L'Ecuyer and Buist 2005, L'Ecuyer 2004b). It was also found empirically in L'Ecuyer and Buist (2006) that it was best to generate service times and patience times for *all* customers, even when it was not needed; this gave a better synchronization. For comparison, if patience times were generated only for the customers who did not start their service immediately upon arrival, the variance was about 40 times larger! Also, if a single stream was used for all types of random numbers (with no attempt to maintain synchronization), the variance was about 400 times larger (which is still better than independent random numbers by a factor of 75).

Another important situation where CRNs are crucial is for solving an optimization problem where the objective function and/or the constraints contain mathematical expectations that cannot be computed exactly for each setting of the decision variables, but can only estimated by simulation. Performing independent simulations to compare the performance at different settings of interest is typically much too noisy to be effective. One class of methods, called *sample-average optimization*, operates as follows. Suppose (conceptually) that we simulate the model n times at all possible parameter settings (which is usually an infinite set), using CRNs across the different parameter settings, and that we estimate all expectations by the sample average over these n runs. After fixing all the (common) random numbers of the n runs, the sample averages become deterministic functions of the decision variables, which means that we have obtained a *deterministic* optimization problem. This *sample-average problem* can be solved (at least in principle) by our favorite optimization method. To evaluate the sample-average functions (and perhaps their gradients) at any given setting, we must run the simulation at that setting, using well-synchronized CRNs across these settings. Under certain conditions on the model, one can prove that both the optimal value and the optimal solution of the sample-average problem converge to those of the original problem when $n \to \infty$. This convergence can actually be characterized by central-limit theorems and other useful properties (Rubinstein and Shapiro 1993, Ruszczynski and Shapiro 2003).

This type of approach has been used successfully for example to optimize the staffing and scheduling of agents in a large call center with multiple types of agents (Ceşik and L'Ecuyer 2006, Avramidis et al. 2007). In that setting there can be thousands of integer-valued decision variables, and the sample-average problem is solved via linear or integer programming. This requires simulating the system at millions of configurations, with CRNs. There is no chance that doing this with independent random numbers would provide a good solution in reasonable time.

IMPORTANCE SAMPLING

Importance sampling (IS) consists in changing the probability laws of the input random variables of a simulation, usually with the aim of concentrating the sampling effort in the most important areas of the sample space. It is the primary way of dealing with *rare event simulation*, i.e., situations where certain types of rare events have an important impact on the performance of interest (Juneja and Shahabuddin 2006). To be specific, suppose we want to estimate $\mu = \mathbb{E}[h(\mathbf{Y})]$ for some function $h : \mathbb{R}^d \to \mathbb{R}$, where $\mathbf{Y}$ is a continuous random vector with density $\pi(\mathbf{y})$ over the d-dimensional real space $\mathbb{R}^d$. If g is any another density

such that $g(\mathbf{y}) > 0$ whenever $h(\mathbf{y})\pi(\mathbf{y}) \neq 0$, then we can write

$$\begin{aligned}\mu &= \mathbb{E}_\pi[h(\mathbf{Y})] = \int_{\mathbb{R}^d} h(\mathbf{y})\pi(\mathbf{y})d\mathbf{y} \\ &= \int_{\mathbb{R}^d} [h(\mathbf{y})\pi(\mathbf{y})/g(\mathbf{y})]g(\mathbf{y})d\mathbf{y} \\ &= \mathbb{E}_g[h(\mathbf{Y})\pi(\mathbf{Y})/g(\mathbf{Y})] \qquad (1)\end{aligned}$$

where $\mathbb{E}_\pi$ denotes the mathematical expectation under the density π and similarly for g. This means that if $\mathbf{Y}$ is generated from density g, then

$$X_{\text{is}} = h(\mathbf{Y})\pi(\mathbf{Y})/g(\mathbf{Y}) \qquad (2)$$

is an unbiased estimator of μ. Here, the original estimator $X = h(\mathbf{Y})$ is multiplied by the *likelihood ratio* $L(\mathbf{Y}) = \pi(\mathbf{Y})/g(\mathbf{Y})$. For discrete random variables, just replace the densities by probability mass functions.

If h cannot take negative values, it is always possible (in theory) to reduce the variance to zero by taking $g(\mathbf{y})$ proportional to $h(\mathbf{y})\pi(\mathbf{y})$. It turns out that the right proportionality constant, so that g integrates to 1, is $1/\mu$. Then the estimator is always equal to μ. Implementing this in practice is usually more difficult than estimating μ in the first place, but it provides a guideline on how to select a g that could reduce the variance, by approximating the optimal one. In a nutshell, we want to inflate the density by a factor that is approximately proportional to $h(\mathbf{y})$. As a special case, when estimating the probability of a rare event, the optimal g is the original density conditional on the occurence of the rare event. This idea generalizes to the simulation of Markov chains in general, with arbitrary state spaces. This is studied in another paper by L'Ecuyer and Tuffin (2007), in these proceedings.

There are several types of real-life situations where the probability that an event of interest (that has an impact on the performance) occurs in the simulation is extremely small, for example smaller than 10^{-9} or even less. Then, with standard Monte Carlo, we would need to perform an excessively large number of runs to obtain any kind of meaningful estimator. In this rare-event simulation context, the main viable approaches are IS, together with another class of methods called *splitting* (Ermakov and Melas 1995, Glasserman et al. 1999, Garvels 2000, L'Ecuyer et al. 2007). Concrete examples where arbitrary small probabilities can be estimated very easily under the appropriate change of density can be found in Glasserman (2004), Juneja and Shahabuddin (2006), L'Ecuyer (2008), elsewhere in these proceedings, and in many other places. The variance reduction factor can be arbitrary large.

For a specific numerical illustration, suppose an insurance firm receives money continuously at rate $c > 0$, and receives claims (that they must pay) according to a Poisson process with rate $\lambda > 0$. The claim sizes are i.i.d. random variables with density f. The firm starts with an initial amount $R(0)$, and we want to estimate the probability that its amount in hand (the reserve) eventually becomes negative. This is the *ruin probability* and it should be very small. Besides the fact that ruin rarely occurs, another important difficulty with standard Monte Carlo is that we can never be sure that ruin will not occur unless we simulate over an infinite time horizon. This is of course totally unpractical.

It turns out that a good IS scheme in this case is to replace the density $f(x)$ of C_j by $f_\theta(x) = f(x)e^{\theta x}/M_f(\theta)$ and to increase the rate of the Poisson process to $\lambda_\theta = \lambda + \theta c$, where $M_f(\theta) = \int_{-\infty}^{\infty} f(x)e^{\theta x}dx$ and θ is the largest solution to the equation $M_f(\theta) = (\lambda + \theta c)/\lambda$. Under this change of densities, it turns out that the ruin occurs with probability 1, and the corresponding (unbiased) estimator of μ simplifies to

$$X_{\text{is}} = \exp[\theta(R - R(0))],$$

where R is the (negative) reserve when the ruin occurs.

Suppose, for example, that $R(0) = 200$, $\lambda = 1$, the claim sizes are exponential with mean 2, and $c = 3$. Then the ruin probability is $\mu \approx 2.2 \times 10^{-15}$. Even if we were able to simulate over an infinite horizon at unit cost, estimating such a small probability by standard Monte Carlo would be too costly. The variance per run would be $\mu(1-\mu) \approx \mu$, so the relative error would be $1/\sqrt{\mu}$, which increases to infinity when $\mu \to 0$. With the IS scheme described above, the variance is reduced to 6.3×10^{-31}, a reduction by a factor of about 3×10^{15}. If we take $c = 10$ instead, then we have $\mu \approx 3.6 \times 10^{-36}$ whereas the variance with IS is 2.3×10^{-71}, a reduction by a factor of about 1.5×10^{35} compared with standard Monte Carlo. Estimating this small probability of 3.6×10^{-36} with 10% relative error would require a sample size of approximately $n = 2.8 \times 10^{37}$ without IS, and $n = 183$ with IS.

CONTROL VARIATES

With *control variates* (CV), we exploit auxiliary information to improve our estimate of the average performance of interest. Suppose X is the basic estimator and $\mathbf{C} = (C^{(1)}, \ldots, C^{(q)})^{\mathrm{t}}$ is a q-dimensional vector of control variates, correlated with X, with known expectation $\mathbb{E}[\mathbf{C}] = \boldsymbol{\nu} = (\nu^{(1)}, \ldots, \nu^{(q)})^{\mathrm{t}}$ (the $^{\mathrm{t}}$ is for vector and matrix

transposition). The *controlled* estimator is

$$X_c = X - \beta^t(\mathbf{C} - \nu),$$

where $\beta = (\beta_1, \ldots, \beta_q)^t$ is a vector of constants. We have $\mathbb{E}[X_c] = \mathbb{E}[X] = \mu$.

Let $\Sigma_C = \mathrm{Cov}[\mathbf{C}]$, a matrix whose element (i, j) is $\mathrm{Cov}[C^{(i)}, C^{(j)}]$, and let $\Sigma_{CX} = (\mathrm{Cov}(X, C^{(1)}), \ldots, \mathrm{Cov}(X, C^{(q)}))^t$. We assume that $\mathrm{Var}[X] = \sigma^2 < \infty$, Σ_C and Σ_{CX} are finite, and Σ_C is positive definite. Then, we have

$$\mathrm{Var}[X_c] = \mathrm{Var}[X] + \beta^t \Sigma_C \beta - 2\beta^t \Sigma_{CX}.$$

This variance is minimized by taking

$$\beta = \beta^* = \Sigma_C^{-1} \Sigma_{CX},$$

and this gives

$$\mathrm{Var}[X_c] \;=\; (1 - R_{CX}^2)\mathrm{Var}[X]$$

where

$$R_{CX}^2 = \frac{\Sigma_{CX}^t \Sigma_C^{-1} \Sigma_{CX}}{\mathrm{Var}[X]}.$$

This squared correlation coefficient indicates by what fraction the variance is reduced when we take the optimal β. With a perfect correlation of ± 1, the variance is reduced to zero. This means that the variance reduction factor can be arbitrary large. The variance and covariance terms that define β^* are usually estimated from the same simulation runs; the CV scheme is then equivalent to a linear regression. For theoretical analysis and other variants and recommendations, one can consult Nelson (1990), Glynn and Szechtman (2002), Glynn (1994), for example. A related class of methods called *moment matching*, widely used in finance applications (Boyle et al. 1997), turns out to be asymptotically dominated by linear CVs.

For a concrete illustration, we consider the pricing of an Asian call option on a single asset whose price evolves as a geometric Brownian motion (the exponential of a Brownian process) $\{S(t), t \geq 0\}$. The process is observed at the fixed times $0 = t_0 < t_1 < \cdots < t_c = T$. We want to estimate the option value given by $\mathbb{E}[X]$ where

$$X = e^{-rT} \max\left(0, \frac{1}{t}\sum_{j=1}^{c} S(t_j) - K\right),$$

and r and K are given positive constants. This can be done by generating n independent realizations of the same path, and averaging the n realizations of X. Interestingly, if we replace the arithmetic average in the definition of X by a geometric average, we obtain a random variable

$$C = e^{-rT} \max\left(0, \prod_{j=1}^{c} (S(t_j))^{1/c} - K\right),$$

whose expectation $\nu = \mathbb{E}[C]$ can be computed exactly by exploiting the famous Black-Scholes formula. Kemna and Vorst (1990) proposed using this C as a CV. Lemieux and L'Ecuyer (1998) have observed variance reductions by factors of up to a million for some examples, with this technique.

CONDITIONAL MONTE CARLO

The idea is to *hide information* and replace the basic estimator X by its expectation conditional on the information that remains available. Suppose $\mathscr{G}$ is a sigma-field that contains not enough information to compute the realization of X, but enough to compute its expectation conditional on $\mathscr{G}$. The *conditional Monte Carlo* (CMC) estimator is then

$$X_e = \mathbb{E}[X \mid \mathscr{G}].$$

We have $\mathbb{E}[X_e] = \mathbb{E}[\mathbb{E}[X \mid \mathscr{G}]] = \mathbb{E}[X]$, so this estimator is unbiased, and an application of the Rao-Blackwell theorem gives

$$\mathrm{Var}[X_e] \;=\; \mathrm{Var}[X] - \mathbb{E}[\mathrm{Var}[X \mid \mathscr{G}]] \;\leq\; \mathrm{Var}[X].$$

The choice of $\mathscr{G}$ is a matter of compromise. The less information it contains, the more the variance is reduced. But if it contains too little information, X_e becomes too difficult to compute.

In many situations, X is defined as a sum of random variables, say $C_1, C_2, \ldots$, and it may be much more convenient to apply CMC to each C_j separately, with a different sigma field $\mathscr{G}_j$. This *extended* CMC methodology is no longer guaranteed to reduce the variance, but it can be very convenient and effective in certain cases.

CMC by itself rarely reduces the variance by a huge factor such as those mentioned in the previous examples. Typical gains are quite modest. However, CMC can make a big hit when combined with a finite difference estimator with CRNs. There are indeed many situations where Proposition 1 (ii) does not apply because $X(\theta)$ is not continuous in θ, but where $X_e(\theta) = \mathbb{E}[X(\theta) \mid \mathscr{G}]$ is continuous in θ for an appropriate choice of conditioning $\mathscr{G}$. The role of CMC in that context is to boost the efficiency of the derivative estimator, and it can make a huge difference, as can be seen by comparing Proposition 1 (ii) and (iii). Several examples

of this are given by L'Ecuyer and Perron (1994) and Fu and Hu (1997). A frequent situation where this applies is when the estimator X involves indicator functions.

As an illustration, suppose that in the call center example discussed earlier, we want to estimate the expected number of abandonments in a day, and the derivative of this expectation with respect to a service time parameter θ. The basic estimator $X(\theta)$ here is just the number of abandonments in the day. It is piecewise constant in θ for any fixed set of underlying uniform random numbers. Suppose now that we erase from the simulation's sample path all the calls that have abandoned and all traces of these calls. Let $\mathscr{G}$ be the information that remains. If the calls arrive according to a (non-stationary) Poisson process and if the patience times are all exponential, then it is possible to compute the expected number of abandonments conditional on $\mathscr{G}$. The idea is to multiply the arrival rate at time t by the probability that a call arriving at time t would have abandoned before starting its service, and integrate this with respect to t for the entire day. Under mild conditions, this conditional expectation would be continuous with respect to θ, so it would provide a much more accurate derivative estimator.

GENERALIZED ANTITHETIC VARIATES AND RANDOMIZED QUASI-MONTE CARLO

There two classes of techniques are essentially the same idea viewed from slightly different angles. We estimate the expectation of X by the average of k realizations of X, say $X^{(1)},\ldots,X^{(k)}$, each having the same distribution as X, and we want to induce *negative dependence* between these k copies of X. The (unbiased) estimator based on *generalized antithetic variates* (GAV) is

$$X_{\mathrm{a}} = \frac{1}{k}\sum_{i=1}^{k} X^{(i)}$$

with variance

$$\begin{aligned}\mathrm{Var}[X_{\mathrm{a}}] &= \frac{1}{k^2}\sum_{j=1}^{k}\sum_{\ell=1}^{k}\mathrm{Cov}[X^{(j)},X^{(\ell)}] \\ &= \frac{\mathrm{Var}[X]}{k} + \frac{2}{k^2}\sum_{j<\ell}\mathrm{Cov}[X^{(j)},X^{(\ell)}].\end{aligned}$$

The variance is reduced in comparison with independent runs if and only if the last sum is negative. The goal is to make it as negative as possible. One way of doing this is via antithetic pairs of random variables (with $k = 2$). Another way is Latin hypercube sampling. But these methods are generally dominated by randomized quasi-Monte Carlo (RQMC).

RQMC is based on the following intuition. Suppose that generating X requires s independent uniform random variables over $(0,1)$, so that its expectation can be written as the integral of some function f over the s-dimensional unit hypercube. Ordinary Monte Carlo then corresponds to generating independent random points in this hypercube, evaluating f at each point, and taking the average. The idea of RQMC is to start with a set $P_k = \{\mathbf{u}_0,\ldots,\mathbf{u}_{k-1}\}$ of k points that cover the unit hypercube $[0,1)^s$ in a very uniform way, and randomize P_k so that after the randomization:

(a) P_k retains its high uniformity when taken as a set and
(b) each individual point of P_k has the uniform distribution over $[0,1)^s$.

If $X^{(i)}$ represents the value taken by f at the ith randomized point, then we are back to the GAV setting. To estimate the variance and compute confidence intervals, it is usually necessary to repeat the randomization m times, independently, for some positive integer m, to obtains m independent copies of X_{a}. The variance is then estimated from the sample variance of these m independent copies.

Specific ways of constructing highly-uniform point sets P_k are detailed in Niederreiter (1992), for example. They are usually known under the name of *low-discrepancy* or *quasi-Monte Carlo* point sets. The main classes of constructions are digital nets and lattice rules (Niederreiter 1992, L'Ecuyer and Lemieux 2002, Glasserman 2004). Randomization methods that satisfy (a) and (b) above are examined in L'Ecuyer and Lemieux (2002), for example. Software implementations are available in SSJ (L'Ecuyer 2004b). Under the assumption that the integrand f is smooth enough, these methods provide estimators whose *worst-case* square error converge as $O(n^{-2}(\log n)^{2s})$, which is asymptotically better than the $O(n^{-1})$ convergence rate for the *mean* square error in the standard Monte Carlo method. One could rightfully argue about the practical significance of this asymptotic result when s is larger than 10 or so. But there are many important applications, e.g., in computational finance, computer graphics, and computational statistics, where RQMC is really effective (Keller 2006, L'Ecuyer 2004a; 2008).

What happens is that even if s is large, it is often possible to redefine the function f in a way that f is well-approximated by a sum of low-dimensional functions, and it suffices that these low-dimensional functions are integrated with good accuracy by the RQMC method. Concrete examples of this can be found in L'Ecuyer (2004a), Glasserman (2004), for example. In numerical examples of option pricing, L'Ecuyer (2004a) obtains variance reductions by factors of up one million in certain cases.

ACKNOWLEDGMENTS

This work has been supported by Grant OGP-0110050 and a Canada Research Chair to the author.

REFERENCES

Avramidis A.N.; Gendreau M.; L'Ecuyer P.; and Pisacane O., 2007. *Simulation-Based Optimization of Agent Scheduling in Multiskill Call Centers*. In *Proceedings of the 2007 Industrial Simulation Conference*. Eurosis, 255–263.

Boyle P.; Broadie M.; and Glasserman P., 1997. *Monte Carlo methods for Security Pricing*. *Journal of Economic Dynamics and Control*, 21, 1267–1321.

Ceşik M.T. and L'Ecuyer P., 2006. *Staffing Multiskill Call Centers via Linear Programming and Simulation*. *Management Science*. To appear.

Ermakov S.M. and Melas V.B., 1995. *Design and Analysis of Simulation Experiments*. Kluwer Academic, Dordrecht, The Netherlands.

Fishman G.S., 1996. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Series in Operations Research. Springer-Verlag, New York, NY.

Fréchet M., 1951. *Sur les Tableaux de Corrélation dont les Marges sont Données*. *Annales de L'Université de Lyon, Sciences Mathématiques et Astronomie*, 14, 53–77.

Fu M. and Hu J.Q., 1997. *Conditional Monte Carlo*. Kluwer Academic, Boston.

Fu M.C., 2006. *Gradient Estimation*. In S.G. Henderson and B.L. Nelson (Eds.), *Simulation*, Elsevier, Amsterdam, The Netherlands, Handbooks in Operations Research and Management Science. 575–616. Chapter 19.

Garvels M.J.J., 2000. *The Splitting Method in Rare Event Simulation*. Ph.D. thesis, Faculty of mathematical Science, University of Twente, The Netherlands.

Glasserman P., 1991. *Gradient Estimation Via Perturbation Analysis*. Kluwer Academic, Norwell, MA.

Glasserman P., 2004. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glasserman P.; Heidelberger P.; Shahabuddin P.; and Zajic T., 1999. *Multilevel Splitting for Estimating Rare Event Probabilities*. *Operations Research*, 47, no. 4, 585–600.

Glynn P.W., 1994. *Efficiency Improvement Techniques*. *Annals of Operations Research*, 53, 175–197.

Glynn P.W. and Szechtman R., 2002. *Some New Perspectives on the Method of Control Variates*. In K.T. Fang; F.J. Hickernell; and H. Niederreiter (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, 27–49.

Glynn P.W. and Whitt W., 1992. *The Asymptotic Efficiency of Simulation Estimators*. *Operations Research*, 40, 505–520.

Juneja S. and Shahabuddin P., 2006. *Rare Event Simulation Techniques: An Introduction and Recent Advances*. In S.G. Henderson and B.L. Nelson (Eds.), *Simulation*, Elsevier, Amsterdam, The Netherlands, Handbooks in Operations Research and Management Science. 291–350. Chapter 11.

Keller A., 2006. *Myths of Computer Graphics*. In H. Niederreiter and D. Talay (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2004*. 217–243.

Kemna A.G.Z. and Vorst A.C.F., 1990. *A Pricing Method for Options Based on Average Asset Values*. *Journal of Banking and Finance*, 14, 113–129.

L'Ecuyer P., 1990. *A Unified View of the IPA, SF, and LR Gradient Estimation Techniques*. *Management Science*, 36, no. 11, 1364–1383.

L'Ecuyer P., 1991. *An Overview of Derivative Estimation*. In B.L. Nelson; W.D. Kelton; and G.M. Clark (Eds.), *Proceedings of the 1991 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 207–217.

L'Ecuyer P., 1994. *Efficiency Improvement via Variance Reduction*. In *Proceedings of the 1994 Winter Simulation Conference*. IEEE Press, 122–132.

L'Ecuyer P., 2004a. *Quasi-Monte Carlo Methods in Finance*. In R.G. Ingalls; M.D. Rossetti; J.S. Smith; and B.A. Peters (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, New Jersey.

L'Ecuyer P., 2004b. *SSJ: A Java Library for Stochastic Simulation*. Software user's guide, Available at `http://www.iro.umontreal.ca/~lecuyer`.

L'Ecuyer P., 2008. *Stochastic Simulation*. Notes for a graduate simulation course, textbook in preparation.

L'Ecuyer P. and Buist E., 2005. *Simulation in Java with SSJ*. In *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, 611–620.

L'Ecuyer P. and Buist E., 2006. *Variance Reduction in the Simulation of Call Centers*. In *Proceedings of the 2006 Winter Simulation Conference*. IEEE Press, 604–613.

L'Ecuyer P.; Demers V.; and Tuffin B., 2007. *Rare-Events, Splitting, and Quasi-Monte Carlo*. *ACM Transactions on Modeling and Computer Simulation*, 17, no. 2, Article 9.

L'Ecuyer P. and Lemieux C., 2002. *Recent Advances in Randomized Quasi-Monte Carlo Methods*. In M. Dror; P. L'Ecuyer; and F. Szidarovszky (Eds.), *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, Kluwer Academic, Boston. 419–474.

L'Ecuyer P. and Perron G., 1994. *On the Convergence Rates of IPA and FDC Derivative Estimators*. *Operations Research*, 42, no. 4, 643–656.

L'Ecuyer P. and Yin G., 1998. *Budget-Dependent Convergence Rate for Stochastic Approximation*. *SIAM Journal on Optimization*, 8, no. 1, 217–247.

Lemieux C. and L'Ecuyer P., 1998. *Efficiency Improvement by Lattice Rules for Pricing Asian Options*. In D.J. Medeiros; E.F. Watson; J.S. Carson; and M.S. Manivannan (Eds.), *Proceedings of the 1998 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 579–586.

Nelson B.L., 1990. *Control-Variate Remedies*. *Operations Research*, 38, 974–992.

Niederreiter H., 1992. *Random Number Generation and Quasi-Monte Carlo Methods*, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia, PA.

Rubinstein R.Y. and Shapiro A., 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley, New York.

Ruszczynski A. and Shapiro A. (Eds.), 2003. *Stochastic Programming*. Handbooks in Operations Research and Management Science. Elsevier, Amsterdam, The Netherlands.

AUTHOR BIOGRAPHY

PIERRE L'ECUYER is Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada, and a member of the GERAD and CIRRELT research centers. He holds the Canada Research Chair in Stochastic Simulation and Optimization. His main research interests are random number generation, quasi-Monte Carlo methods, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He is currently Associate/Area Editor for ACM Transactions on Modeling and Computer Simulation, ACM Transactions on Mathematical Software, Statistical Computing, International Transactions in Operational Research, The Open Applied Mathematics Journal, and Cryptography and Communications. He obtained the E. W. R. Steacie fellowship in 1995-97, a Killam fellowship in 2001-03, and he was elected INFORMS Fellow in 2006. His recent research articles are available on-line from his web page: `http://www.iro.umontreal.ca/~lecuyer`.

ABOUT THE PREDICTABILITY AND COMPLEXITY OF COMPLEX SYSTEMS

Renate Sitte
Griffith University
SEET/ICT, Gold Coast Campus
Griffith University, QLD 4222, Australia
E-mail: r.sitte@griffith.edu.au

KEYWORDS
Complex systems modeling and simulation

ABSTRACT

Understanding and prediction are the ultimate goals in Modeling and Simulation. In this endeavor, we are faced with ever-increasing complexity. This works twofold with ever more complex models and modeling techniques mapping to the increasingly complex phenomena that we are trying to simulate. With steadily raising computing capability, we are able to incorporate more complexity into our models and simulations. In return, this has had an impact on our thinking paradigm and capability for abstraction and modeling techniques. So, when are things complex? Our thorough analysis has revealed that complexity can mean very different things to different people, contexts and environments. In many cases, it refers to an aspect or part of the system under study, in others the magnitude and variety of the system itself. To some "complexity" rather means the unknown: that what cannot be easily understood, modeled or predicted with current knowledge. In this keynote address, I will present a unifying and systematic approach to complexity, to bring some clarity into the unknown, and a step further towards predictability. We will be looking at different types of complexity, ranging from simple structural and functional complexity, to those highly complex systems that change (but not necessarily evolve) through their own dynamics. With this in mind, we will be able to identify what can be predicted and what cannot. By knowing it, we will be in a better position to find the unknown, which is the most important step towards the prediction of complex systems.

INTRODUCTION

Since ancient times, one of human's endeavors is taking control of what happens around us, and affects us in one or another way. To take control or even influence our surroundings, we have to know where and what will have an effect. This means that we have to be able to manipulate all or part of what we want to control. To do this we must have a mental mapping, a model of its components and their functionality, and this in turn requires some understanding. This works fine so long things are not getting too complex. The human brain is limited to handle about 2.5 bits of information at a time (known as the 7±2 "chunks"), for anything more, we need memorizing aids such as diagrams, itemized text, etc. For larger projects, this can grow quickly into many pages, folders, even boxes of information storing papers or other media. With the invention of modern computers, we have been able to solve ever-growing large and complex problems. So too is the complexity of the problems to solve. Since J. Forrester's World models in the early seventies, the past decades have seen cycles of ever-increasing complexity and computing power, pulling each other further. We have seen astonishing results. The word "complexity has become a fashion statement in the research communities. Everything is "complex" now: it sounds good, adds magic and importance, and has the potential to attract funding. But, there is more to it than just fashion. There is a strong reason behind being able to understand and deal with complexity. Here is an example. Refraction through another medium is being modeled by measuring the incident angle with respect to the vertical, and then using the refraction index of the medium and exit angle in a simple equation that we learnt in high school. Ancient Arabs knew that the medium had different refraction (indices) and tried to set up a model, but they failed because their model was highly complex. In fact, the model became very complex because they measured the incident angle with respect to the horizontal! So, a rotation of 90^{o} made all the difference between a very simple and a complex model.

During humankind's different stages of intellectual evolution, different schools, insights and paradigms have emerged, such as the development of mathematics, scientific method and systems thinking that lead in one or another way to our current state of the art. The success of any of these approaches is due to a consistent clearly defined underlying definitions and theory. While we are still far away from a widely applicable definition of complexity, this keynote presentation is to bring awareness on how to tackle complex systems as good strategies, as a practical approach rather than a prescriptive general technique.

WHEN ARE THINGS COMPLEX?

We would all agree in finding that systems are complex when they undergo changes in structure, changes in functioning, when they have adaptive feedback, are evolving, or simply when they have a large number of components or relations. One of the hurdles in defining a

theory of complexity, and with it developing a fundamental, helpful approach is that there is no uniformity in the meaning of complexity. Complexity means different things to different people. In a large literature review, we have found that sometimes it refers to an aspect or part of a system, other times to the magnitude and variety of the system. To others it meant the inexplicable or the emergence of unexplained behavior. To yet another group it means chaotic systems, or the difficult, the unknown: that what cannot be easily understood, modeled or predicted with current knowledge. We have to move away from the subjective to an objective notion of complexity. We have found that perhaps the strongest characteristics of a complex system are that they have:

- Intricate interdependencies among their many parts
- Many variables operating simultaneously
- Generally nonlinear, this can include butterfly effect
- Cause and effect are not close in time and space
- Intuitive interventions do not produce the expected outcome
- Reductionist analysis fails (misleading)

We have to see the domain of complexity in a spatial and temporal scale. This is very helpful, but does not necessarily mean that complexity grows with time or space, at least not always. We could find as many examples as counterexamples to it. Thirty years ago, programs such as Forrester's World programs were stretching the capability of computers, but manybody interaction is still a problem with current computer power. Is this complex? No, it is limited computing capability. The models are there. There is a paradox in complex systems. For example the well-known Mandelbrot patterns are generated with very simple rules, but the patterns they generate are quite complex. Another classic example of the paradox is a Braitenberg (Braitenberg 1986) vehicle (with two sensors and two motors) that moves around in an apparent complex trajectory, but in reality it is a rather simple phenomenon, because as light falls on one or another light sensor, the vehicle responds. It is a paradox between cause and effect, but it is not a general rule.This paradox shows that in a complex system not all is complex. Therefore it is important to identify in what, where the complexity is located. This is a very important step in dealing with a complex system. We have to be aware that there are different types of complexity. We can distinguish between the following fundamental types of complexity:

- Functional
  - Single or multifunctional components …
- Structural
  - Dimensionality
  - Networks
  - Hierarchy
  - Levels – depth – breadth

- Topological
  - Connectivity
  - Relation
  - Number of relations
  - Direction of relations
- Algorithmic
- Architecture
  - Series, parallel, cascade, recursion
- Extent of …

This list could be expanded to include more types and sub types. For example, one could argue whether we would include music into the structural types or into an additional class of – say a sensorial complexity. This would lead to an argument of where to draw the boundaries between objectivity and subjectivity. We leave this matter for the moment, as it will be addressed in quantifying complexity. In any of these types, we have to consider the extent of repetition and or parallelism, and the extent of homomorphism (same structure) and isomorphisms (same functionality) that may occur. Here are some examples for extent of repetition: (a) redundant components in safety critical applications, say an aircraft or medical software, and (b) to enhance efficiency in a grid of computers or turbines or windmills for electricity generation.

We can also see that some of these types have affinities with each other. The diagram in Figure 1 shows relations between these fundamental types of complexity have.

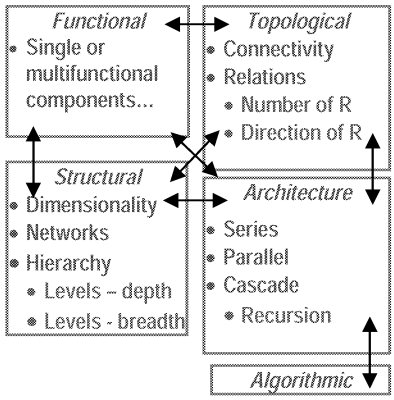


Figure 1 Fundamental types of complexity and their affinities

A classification is rarely useful on its own. A complex system is likely to have more than one type of complexity. The types of complexity do not map to each other in equal proportions or weight. The way in which this taxonomy approach is helpful is by identifying where the complexity is bundled. Is it because the network is complex? Is it because it is a highly multi-functional system? To identify this allows taking a better approach in handling the complex system, by using techniques that are more suitable.

If we know more about the nature of complexity, we are able to recognize patterns, derive heuristics and given time, develop a general complexity theory in its own right. A word of caution: the concept of complexity theory as used in computing (the "big O"), refers to algorithmic complexity. Its applicable to other complex systems is very limited.

MODELLING COMPLEXITY

The ultimate purpose of modeling is prediction. If we can understand and model a process, then we are able to anticipate its behavior. This in turn enables to take rational decisions and to change the behavior of that process, if possible at all. There is no specific environment for modeling complex systems. In complex systems modeling, this often implies using a range of different models, requiring different software tools. Conventional commercial software and toolboxes can be a substantial help, in particular because of their efficient algorithms and user-friendly interfaces. They are tested, which is sometimes an underestimated advantage. However, there is no guarantee that the simulations are truthful just because the code works correctly if the models or the results are not validated. As always in modeling and simulation, the choice of software must suit the models. Regrettably enough, too many times one can find work with wrongly applied models or inappropriate tools in the scientific literature. Some popular techniques that are used in modeling complex systems (in increasing difficulty) are dynamic systems (Matlab/Simulink); Artificial Neural Networks, including Self Organizing Maps (SOM) for functions and clustering; Chaos, and Quantum Computing

Before we move on to how to tackle complexity, let us refresh the vectorial model for a problem, because it offers some intuition to what comes later. Imagine that we have 300 dollars and have to pay 500 dollars. This means that we have a problem of 200 dollars. In this case, we are in one single dimension, the dollars. We could add time or interests as other dimensions. In the vectorial model of a problem, we have a current state position (represented by a vector) and a desired position (another vector). The length and direction of the vectors represent quantities that characterize these states. We consider the problem as the necessary transition to move from the current state position, to the desired state position. The necessary steps to achieve the transition are then the solution to the problem. This notion is shown in Figure 2 as a 3D vectorial representation. It can be extended to a multidimensional space that is typical in dynamic systems.

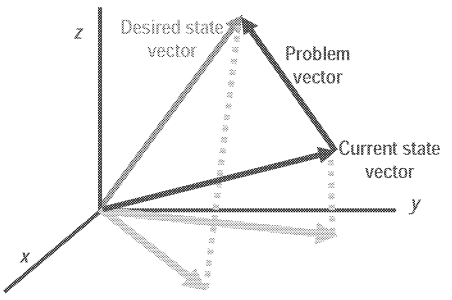


Figure 2 Vectorial model for a problem in a 3-dim. space as the transition from a *current state* to a *desired state*. The model can be extended to n-dim.

To tackle complexity in a systematic way requires an unambiguous definition. As mentioned earlier, there is no consensus about what complexity is because it means different things to different people. A definition is necessary to be able to measure complexity; otherwise, we do not know what to measure. We need to measure and quantify complexity to be able to assess complexity, that is the degree of complexity in a process or its representation in a model. Only when we can assess complexity quantitatively are we able to simplify and reduce complexity either in the model or in the process itself, else we do not know what to simplify. This rationale is schematized in Figure 3.

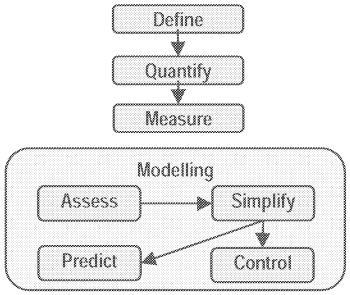


Figure 3 Rationale for a complexity definition and quantification

MEASURING COMPLEXITY

If we can express complexity as a quantity, we can distinguish unambiguously between complex and simple, or to a finer scale, between highly complex, and relatively low complexity. Earlier we have seen a number of rather static complexities: entities, relations amongst them, hierarchy depth, etc. We are now moving into complexity of dynamics, as a continuously changing entity. We do this by bearing the problem vector in mind. It is the quantification in this model that makes it intuitive. If we can describe the complexity in terms of algorithmic operations, then we can find ways of moving from complex to simple. It is therefore that an operational complexity should be sought.

In a major work and milestone, Hazen and his collaborators (Hazen et al. 2007) have developed a definition for system complexity. The authors define *Functional Information* as the fraction of all possible configurations that achieve specified degree of function. The notion of Functional Information is exemplified by a sequence of letters that is conceptually familiar: The four different nucleotides A, U, G, C. A 100-nt RNA strand gives 4100 different possible sequences, but not all of them are functional (i.e. some are dummies and do not play any role). With this simple but powerful definition, and by involving the probability or potential of an entity to perform a function, the authors have derived a set of mathematical models, including sophisticated models for higher dimensional systems.

Basu (Basu 1998) has defined a topological complexity based on their combinatorial and algorithmic properties. A bounding function of the combinatorial complexity is defined using the connectivity of their subsets. The computation (or estimate) of this bounding function poses an important problem in computational geometry i.e. the arrangements of sets of geometric objects.

DEALING WITH COMPLEXITY

In this section, we will discuss a selection of commendable approaches that are helpful in dealing with complexity. Systems Engineering has brought a radical change into the way we approach problems with undeniable success. However, with increasing complexity, one cannot use this approach. Complex systems do not always behave as defined and expected in classical systems theory. The danger is that the traditional breakdown into subsystems, using emergent property as the defining attribute can and does fail in complex systems. In a Systems approach one works on high, or low level models, but without mixing levels. In a complex system, the different levels may not be independent. This is because of interrelations and couplings that can act between the different levels of subsystems, thus violating a fundamental paradigm in systems theory. Based on this, Bar-Yam (Bar-Yam 2003) has found that the systems engineering approach can be used in not-too-complex systems, but recommends for higher complexity systems adopting an evolutionary paradigm for complex systems engineering that involves rapid parallel exploration and a context designed to promote change through competition.

Our first important and wise step is to look at what level of resolution (granularity) our problem to solve lies. An overwhelming amount of data limits modeling detail. The following examples illustrate scale and granularity. The object of interest can be:

Cell → organ → organism → …
Group → organization → society → …

Structure → component → functional unit → …
Cluster → equipment →tool & wider unit →…

from either we could derive
→ dynamics (market, means)

This does not warrant that our solution space will be in the same granularity. Granularity can involve selective detailing. This is because not all needs to be modeled and simulated in equal detail. This comes because mixed modeling approaches can be integrated. One can model in detail those entities that need low-level detail, and all other entities at a high level under given circumstances, despite violating the rules of general systems theory. This practice of "zooming in" is explained in (Sitte 1999) and shown as a diagram in Figure 4. Let us consider that P is a state function of q (they are vectorial functions). Let us consider that this function can be split as the sum of two functions, F_1 and F_2 as shown in the diagram. We can now model some of it in a large resolution (like "zooming in"), and leave all other in a lower resolution. If the original unsplit function P is embedded in another system, then the sum of the high and low-resolution functions F_1 and F_2 has to be carried out before further calculations. For example, we have used this successfully in our dynamic system models on Economy or Energy. We used a rather high level, modular model, with several layers of hierarchy (in Simulink). At some stage, we had to introduce a highly complex function that was only available as a data collection of several decades. We could zoom in on these low-level data by training an artificial Neural Network (NN), and plugging it into our model for future predictions. Here we used the benefit of an NN, which is able to return a weighted function that is able to recognize similar data. The drawback is that the weights or coefficients cannot be interpreted and associated with a cause, as one does in physics. If this interpretation is not necessary, then a NN or other AI technique are suitable zooming in mechanisms within a model of a different, higher resolution. If a modular, e.g. a hierarchical modeling approach is used, then complexity can built in progressively wherever necessary. One has to ensure that (a) the outcome does map to data with the real world, and (b) that the overall result of each subsystem must be maintained and stable. The calls "Calibrate!" and "Validate!" must resound in our ears and actions.

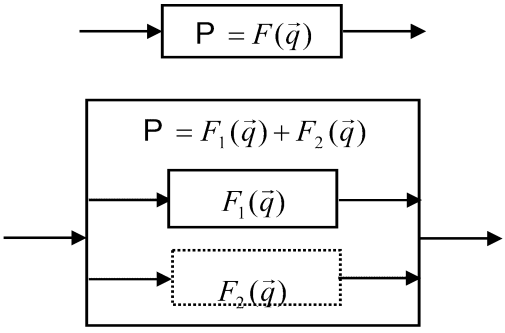


Figure 4 "Zooming-in" applies to hierarchical systems that can be represented as a flow system.

Our second step is to try to ubicate our problem in a spatial and temporal reference. For example, we consider a daily vs. yearly weather forecast, a daily fluctuation vs. yearly stock market trend, metabolism of one cell vs. a whole organism, or a socio-economic system. Figure 5 shows how this applies. This figure is an adaptation from on Wallace (Wallace 2003), which the author has used to show temporal and spatial scales associated with different water related issues in the GWSP-2003 project. For our case on increasing complexity, we could add a third axes with a dimensionality scale. Although this may seem a circular reference, it is not. Dimension is a quantity that can equally be represented on a scaled axis. However, complexity does not necessarily increase at higher dimensions; it actually can be reduced, as we will see later.

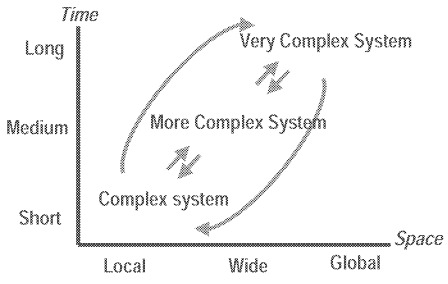


Figure 5 Temporal and spatial complexity increase

The third step is to identify repetition, and if applicable, one should find out whether the repetition makes the system more complex or not. The difference is that if the repetition can be bundled into one functional entity, then it is unlikely to affect the complexity. How much the performance of the system is affected can be used as an indicator, tolerance or threshold (e.g. a percentage).

An old saying tells us to "be wise, linearize". But then complex systems are not linear. Or are they? One main characteristic of complexity is that they are non-linear. But then, not 100% in a complex system is non-linear. This means, that some of the modeling can be linearized. What is important is to find out what can be linearized and what cannot. A sensitivity analysis helps to find out, albeit that it means a substantial effort that can easily run into the combinatorial curse. A sensitivity analysis is always a good approach because it sheds light into what is of primary importance and what is lesser. With a good design of experiments (DOE) it also gives a good indication of where the correlations are stronger. A good question to ask is also, does if follow a Pareto 80/20 rule? More often than not, it will. Again a word of caution: characteristics such as linearity and sensitivity can be limited to specific ranges of values. It is important to be aware of this, and find out whether and where they are.

All simulations are affected with uncertainty. Schneider and Moss from Stanford University insist that "… statistical confidence levels should be attached to even the most complex scientific predictions" (Giles 2002). In dealing with complex systems, it is crucial to know where uncertainty is located. One has to find out not only how much uncertainty is involved, but also where it comes from. This can be represented on a multidimensional axes (radar) graph. Schneider and Moss insist that the uncertainty of at least four sources, namely the theory applied, the observations, the model results, and the consensus should be represented graphically for a clearer information about the source of uncertainty. This allows improving the certainty of the research, or at least being able to pinpoint why the uncertainty arises, and allows comparisons between different approaches.

Reducing Complexity

We are now going to look into ways for reducing complexity. In general complexity reduction should be avoided in favor any other technique, e.g. another level of resolution (granularity). In some cases reducing complexity can be done easily, and in other cases it cannot and should not be done because of the risk involved that non-obvious correlations are violated leading to distorted results. It can only be done with much care. There are no universal recipes for reducing complexity. In the absence of a theory on complexity, at this time perhaps the best way is to look at what has worked, and what not.

Modeling involves abstraction. To what extent can abstraction be done without loss or distortion by over-simplification? The answer lies somewhere in between, that is a compromise between the manageability of the model and its modeling accuracy. A popular approach for reducing complexity is order reduction. This comes from expressing a model as a series expansion. This can be as simple as polynomial series. One uses only the first, second, … n^{th} order terms, neglecting the higher order terms, as they would contribute little or negligible to the results. Caution: Not to be confused with order-predicates in logic.

The literature is full with papers about reducing complexity. Some are good, some are reinventing the wheel or clumsy. In what follows, a selection of examples will be presented. Their didactical purpose should be seen as what the authors have done, rather than the application case itself. The examples should bring the reader into the mental paradigm of reading papers on complexity reduction with a "what can be done" attitude, but not "where can I find a similar work", because it might not apply as a best choice to that researcher's case. This is part of the nature of complexity.

Decomposition and clustering can be powerful techniques in certain cases. A representation (transform) into higher dimensions (in general) will reduce complexity. Sharony (Sharony 1994) has demonstrated the general case, that by switching from a 1D network to a kD network increases the connectivity but reduces the complexity. Sharony used this for routing in a signal processing network. Different ranges of frequencies were assigned to different dimensions to multiplex the signals into different channels.

While increasing the dimensionality to reduce complexity might appear astounding, the effect is that it acts as selection mechanism or filter. The notion is best illustrated graphically as shown in Figure 6. Initially the red and blue balls are mixed. After increasing the dimension, the red balls are separated on one side, and the blue balls are on the other side.

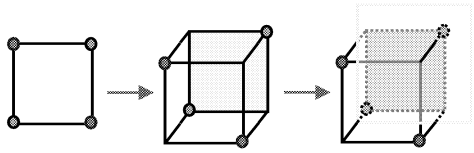


Figure 6 Increasing the dimension will reduce complexity.

Our next topic is on clustering. Clustering appears anywhere: faults and defects in silicon wafers or software, living organisms, cars on a freeway, clouds, chaotic trajectories, etc. Modeling the dynamics of clustering is complex in any discipline. Gibbon & Titi (Gibbon 2005) derived a mathematical model for clustering focusing on an "interface" between a cluster and its surroundings. It is based on the existence of a hierarchy of length scales (correlations, interactions), whose members are in competition, for example an ordered set of correlations or coherence lengths. Depending on parameters, the model yield sets "good" and "bad" regions. Due to Intermittency (gaps) very short length scales crowd into clusters with fractal-like boundaries.

Function transforms can be applied to reduce parameter space. Fast Fourier transforms is commonly applied in mathematics and signal processing, in particular sound and image processing. Modeling fishery dynamics involves estimating time-varying parameters about the growth, stock recruitment, mortality rates, fishing, and others. Each parameter varies with time, and requires one value per time-step. This means that for each time-step the parameter space increases. Subbey (Subbey et al. 2007) represents time-varying parameters using functional representations together with Fast Fourier transforms. The authors have achieved reducing a total of 76 parameters to 36, which is about 47% reduction.

Recognizing when and where a mathematical model or technique can be altered to reduce complexity is good. For decades, chemical plants and oil refineries are known to be complex systems. They involve a considerable variety of products, which in turn involve a huge amount of equipment. They are dangerous places where any mistake puts human lives at risk. The following example is in Operations Research and Parametric Programming. It involves Model Predictive Control (MPC). This is a specialized technique in process control that is used in chemical plants and oil refineries. MP-QP Multi Parametric-Quadratic Programming can be part of MPC. It requires vast number of calculations and data storage, and it is slow. The complexity (number of regions) for an MP-QP solution can be substantially reduced when covering a pre-specified feasible region Rossiter (Rossiter 2006) has found that by splitting and refining the MP-QP algorithm (mathematical model) it is possible to delegate part of the online computation (that slows down the production process) to offline computation. In this way, it is possible to improve speed of online control.

Another example is on decision taking, where strategies are produced as a long-term stable state (wider picture), instead of short-term incidental decisions. In Game theory representations one has typically a dichotomy of decisions (yes/no) in a bounded rationally situation; that is with limited, incomplete or vague info. Mittone (Mittone 2006) has derived a model that incorporates incomplete information into a deterministic frame. To represent all possible moves requires a vast matrix of moves. The model strategy aims for a Nash equilibrium solution, or ideally Pareto equilibrium (no damage to thirds). It then applies an algorithm for maximization of payoff function. After several iterations, eventually the gain stabilizes, yielding an optimal solution. A word of caution: stability may not always be possible.

Reducing complexity can require complex algorithms to achieve success, as demonstrated in the following example. Flexible manufacturing is also considered as a complex system. It involves production with a variety of different products e.g. car models or electronic components, which require alternative combinations of routings. It involves hundreds and millions of routings and their control, whose scheduling and timing is critical. Spiliopoulos and Sofianopoulou (Spiliopoulos and Sofianopoulou 2007) have addressed the Cell (or Cluster) Formation Problem (CFP), i.e. the problem of creating manufacturing cells with minimum interactions. The idea is to decentralize production by grouping the machines into clusters and various part types into part families, then allocate the processing of each part family to a machine cluster. This Eliminates 50% stochastic components. The authors apply first a preprocessing step that develops the tree of possibilities. It steps through the tree examining all combinations of process plans and eliminates equivalent sub-problems. It then applies a bounding technique using an Ant Colony Optimization. The second step is then a nested procedure, whose outer loop examines the cost matrices different combinations, and inner loop applies an algorithm that solves cell-formation problems.

Stability

One has to be aware that stability means two things: (a) the behavior stabilization reached after a disturbance of what is being modeled, for example, an economy going through adaptation after a new taxation scheme, or a forest recovering after a bushfire; and (b) the simulation stabilization time until a model replicates truthfully enough what it is modeling. This depends strongly on the iteration cycles of calculations, on the sensitivity of the model. Feedback systems can be vulnerable to either of them.

A special case is Evolving Systems. They are complex systems, which undergo change in any way. This change can be a structural change, or a control change, or any other, and combinations of them. An example are the morphing, appearance and disappearance of giant companies, political entities (e.g. countries), or large ecosystems. Growth can be internally controlled, or externally. Sometimes these systems can be modeled as dynamic systems, but this is not always the case; in particular when the controlling agent is not clearly identifiable (e.g. a population's movement for democracy and freedom) or the agent changes. A change can be abrupt (e.g. companies) or it may be slow and not readily perceivable. It is interesting to know that non-linear dynamic systems described by three or more equations can suddenly and unpredictably turn into chaotic behavior. Additional input can bring unstable system into a stable mode, an attractor in a chaotic system or into deterministic mode. There is no theory about these systems, but early work by Domingo and Tonella (Domingo and Tonella 2000) explain the requirements for such a theory.

UNSOLVABLE SYSTEMS

In theory, unsolvable systems should not exist. In practice, there are real problems. Here are some examples, a system can be undetermined, it can have infinite solutions, or it can be intractable (i.e. not computable, or not capable to solve with current computer power). The worst of it is that these systems tend to be mischievous. Here is an example: For decades, Engineers have been eagerly researching ways of locomotion of robots, with an impressive range of results. Keeping the balance in rugged terrain, climbing up stairs are still serious challenges for current technologies. Who would not be tempted to try to imitate nature's success? A

spider? Perhaps not an eight but a six-legged robot? Wrong! It is an undetermined system, with 18 degrees of freedom. While this is a problem that has an easy solution, others do not. One has to consider and recognize such situation.

CONCLUSION

With ever-growing complexity of systems to be modeled, there is a strong need for a proper theory of complexity, other than the algorithmic complexity known in computing. The problem is that there is no unanimous consensus as to what complexity is. Several attempts have been made, some are very promising, but a widely applicable theory and practice have not been derived. Quantification is an essential step in modeling to achieve prediction and control of a system. Quantification is also a crucial step in complexity and some complexity quantification models are emerging. In this keynote address, I have presented a unifying and systematic approach to complexity, to bring some clarity into the unknown, and a step further towards predictability. It is presented as an overview and introduction, in particular to the novices on how to deal with complex system as a practical approach. Some practices summarized here are elementary and others are quite ambitious. It happens over and over that the uninitiated researchers make errors, reinvent the wheel or fall into traps. The purpose of this presentation is to offer good advice and a sense on how to avoid pitfalls.

REFERENCES

Bar-Yam Y. 2003. "When Systems Engineering Fails - Toward Complex Systems Engineering", *IEEE International Conference on Systems, Man and Cybernetics*, Volume 2, 5-8 Oct. pp 2021 - 2028

Basu S. 1998. " On the Combinatorial and Topological Complexity of a Single Cell". *Proceedings.39th Annual Symposium on Foundations of Computer Science,* 8-11 Nov, pp 606 - 616

Braitenberg V. 1986. *Vehicles – Experiments in Synthetic Psychology.* MIT Press.

Domingo C., G.Tonella. 2000. "Towards a theory of structural change", *Structural Change and Economic Dynamics,* vol 11, pp 209–225

Gibbon J. D., E. S. Titi. 2005. "Cluster formation in complex multi-scale systems", *Proc Royal Soc.*, vol 461, 3089-3097

Giles J. 2002. "When doubt is a sure thing" (Interview with S.Schneider & R.Moss, Stanford) *Nature*, Vol 418, Issue 6897, 1 August, p476.

Hazen R. M., Griffin P. L., Carothers J.M., Szostak J. W. 2007. *Proceedings of The National Academy of Sciences of The United States of America* vol 104, pp 8574-8581.

Mittone L. 2006. "The reduction of decision complexity: normative policies and the role of information", *Neurocomputing* Volume 69, Issues 16-18, Oct 2456-2460

Rossiter J. A., B. Kouvaritakis, M. Cannon. 2005. "An algorithm for reducing complexity in parametric predictive control", *International Journal of Control* vol:78 iss:18, pp 1511 -1520

Sharony J. 1994. "The Universality of Multi-dimensional Switching Networks", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 6, Dec. pp 602-612

Spiliopoulos K. and S. Sofianopoulou. 2007. "Manufacturing cell design with alternative routings in generalized group technology: reducing the complexity of the solution space", *International Journal of Production Research*, Vol. 45, No. 6, 15 March, pp 1355–1367

Sitte R. 2002. "Zooming-In Modeling Method For Managing Complexity", *International Journal of Modeling and Simulation*, Vol. 21, No. 4

Subbey S., D. Howell, B. Bogstad, M. Åsnes. 2007. "Reducing fisheries model complexity using a finite Fourier series reparameterization", *Fisheries Research* vol 84 pp 390–394

Wallace J.S. 2003. "HELP bridging scales in water science, management and policy", *Proceedings of the International Congress on Modeling and Simulation*, 14-17 July, Townsville, pp 428-433

AUTHOR BIOGRAPHY

RENATE SITTE was born in Vienna, Austria, and lived for longer periods in Venezuela, Sweden and Australia. She obtained her PhD (1995) and MPhil (1993) from Griffith University in Microelectronic Engineering. She received her Systems Engineering degree (Ingeniero de Sistemas) in 1985 from the Universidad de Los Andes in Venezuela. She began her academic career as a faculty staff in 1986 in the early days of the Faculty of Engineering and Information Technology, Griffith University, Australia. She is now an adjunct Associate Professor in SEET/ICT at Griffith University. Her field of research is in modelling and simulation, with over 80 publications. She has supervised many postgraduate students, several in a major project in Virtual Reality Prototyping Micro-electro-mechanical systems (MEMS), which has provided and endless source for research topics.

SIMULA AND 40 YEARS OF OBJECT-ORIENTED PROGRAMMING

Eugene Kindler
University of Ostrava
Department of Informatices and Computers
30. dubna 22, 701 03 Ostrava 1, Czech Republic
FAX: +420 596 120 478
E-mail: ekindler@centrum.cz

KEYWORDS
SIMULA history, Object-oriented programming, Process orientation, Block orientation, Nesting classes.

ABSTRACT

2007 presents a jubilee of 40 years of the object-oriented programming (OOP). In the paper, the importance of computer simulation for discovering OOP is explained, together with the development from process-oriented simulation languages up to SIMULA 67 and behind and above the paradigm called OOP.

INTRODUCTION

For several decades of years, the object-oriented programming is a common paradigm of programming. After having invented the term object-oriented programming (further OOP) in the seventies of the XX century, some non-professional definitions of the meaning of this term were formulated (e.g. "OOP is a programming like in SmallTalk") and when OOP had become a common, famous and effective way of programming its name was irresponsibly dispersed at almost any way of programming (something like objects can be observed anywhere).

Nowadays, there is almost univocal agreement that OOP covers the paradigm of programming where

(OOP1) knowledge is represented as classes that are encapsulations of data and procedures (methods),
(OOP2) the classes can be arbitrarily specialized to subclasses, and
(OOP3) the contents of the procedures belonging to the classes can be changed in subclasses ("virtuality" of the procedures or "late bindings" between their names and their contents).

The principles of OOP were formulated and then implemented before the name OOP was invented; these principles (and many other) were presented at the IFIP Working Conference on Simulation Programming Languages held in Oslo in May 1967 (Buxton, 1968). The authors were Ole-Johan Dahl and Kristen Nygaard (Dahl and Nygaard, 1968), both dying in 2002. The language they proposed was called SIMULA 67; it was slightly refined one year later (Dahl et al., 1968) but the principles presented at the conference were completely preserved.

FROM GPSS TO SIMULA I

Despite the fact that he principles of OOP were presented to the world professional community at a conference on the simulation languages, it may seem peculiar to connect the first roots of OOP with the first language for discrete event simulation, which was GPSS (Gordon, 1961). But it is so: even before orientation to events as entities, GPSS was based upon the idea that the difficulties to understand many discrete event dynamic systems consist in the fact that such a system is composed of elements, each of which being active (in a metaphor: having its proper life), and the parallel lives cause the dynamic complexity. The formal description of such a system leads to

(G1) formal description of the rules of life (further: ***life rules***) for every ***class*** of similarly behaving elements,
(G2) the formal description of the rules can be like that of algorithms, and
(G3) in a monoprocessor-oriented "world viewing", the parallel operation of the lives can be described by means of ***scheduling statements*** that occur is life rules and that cause automatic switching among the lives of the elements when they are performed.

Nowadays, GPSS could be considered as a predecessor of agent paradigm (Kalisz and Flora, 2000). Nevertheless, GPSS was rather limited:

(G4) for every class, the user had to enter the number of its instances,
(G5) neither the classes nor the data declared for them could get mnemonic names, and
(G6) the interaction among the elements was limited to implicit activities like ordering into queues and conflicts in entering the elements with limited capacity.

During the next years, GPSS was gave stimuli to more developed languages. Namely SOL (Knuth and McNealey, 1964) allowed mnemonic naming of the ***attributes*** (data) of class instances and dynamical generating class instances in any number. Nevertheless that was limited so that during performing its life rules, an instance of a class could generate only its copy, but not an instance of another class.

The most developed language of this sort of simulation languages – called ***process oriented simulation languages*** (Dahl, 1966) – was called SIMULA (and later SIMULA I, in order to distinguish it from the first OOP language).

Compared with SOL, it allowed free generating of class instances and had tools, by which the user could explicitly formulate interactions among elements according to his own conception. They were called ***connection statements*** end had form *inspect X do S*, where *X* was any element and *S* a – may be complex – statement. When the life of an element *Y* met such a statement it performed it so that it tried to interpret all names occurring in *S* as attributes of *X* while the names that did not belong to *X* were interpret as referring to attributes of *Y* (Dahl and Nygaard, 1965, 1966).

FROM SIMULA I TO OOP

SIMULA I offered the following principles:

(P1) defining classes of structures containing data and life rules;

(P2) generating any number of "instances" of such classes, each of which had the data introduced for the class and behaved ("lived") according to the life rules introduced for the class;

(P3) the performance of the life rules was switched among the lives of the instances according to the scheduling statements met in the life rules.

In 1966, O.-J. Dahl was invited to present a tutorial (Dahl, 1966) on the discrete event simulation languages at a NATO advanced summer school on programming languages (in the proceedings (Genuys, 1968) of that meeting a reprint of the Dahl's tutorial published). There O.-J. Dahl met results of another lecturer of the same summer school, C. A. R. Hoare, who presented a tutorial on hierarchical data structures (Hoare, 1968). Hoare introduced further principles:

(P4) defining pointers (to data structures) so that each of them was ***qualified*** into a given class;

(P5) introducing pointers as components of data structures;

(P6) introducing ***remote identifying*** (also told ***dot notation***) for expressing "the component *W* of the data structure pointed by *X*" as *X.W*; note that the qualification of *X* enabled to judge directly on the consistence of *X.W* – the class to that *X* was qualified, should have component *W*; if *W* was a pointer an iteration of dot notation was possible, like *X.W.R*;

(P7) enabling to ***specialize*** any class by adding further components, pointers or other; the new class become a ***subclass*** of the source class; any pointer *R* qualified to a class *Q* could refer to an instance of any subclass of *Q* but in a dot notation like *R.D*, *D* was limited to a component defined by *Q* – otherwise *R.D* was inconsistent.

Dahl (then together with his collaborator and co-author of SIMULA I, K. Nygaard) accepted all these principles as possible enriching of SIMULA; in relation to it, he simply

(P8) added that the classes can have life rules, too,

(P9) introduced way how to express specializing the life rules in subclasses, and

(P10) added a possibility that during performing live rules, an instance can generate instances of arbitrary class.

Dahl several times confirmed (orally and in some printed matter as e.g. in (Dahl, 2002), to be due to Hoare and his paper for discovering the principles of the object-oriented programming. Nevertheless, having read the Hoare's paper, one can only judge about the great Dahl's or about his geniality: what Hoare had discovered, Dahl (often in discussions with Nygaard) generalized; the principles (P4)-(P7) were enlarged in the following way:

(P11) procedures could be among the components of the "data" structures;

(P12) dot notation was allowed to be applied for calling the procedures existing components of "data structures",

(P13) virtual procedures were introduced for which their contents could be (re-)declared in subclasses without changing their names.

Instead of "data structures" that became rather crazy since – beside data – it could contain procedures (and luife rules, too), a better term came into use: ***object***.

The conception of the new language (subject of presentation in 1967) was not yet finished by Dahl and Nygaard but already in this state of explication it is possible to say that the principles of OOP have already been involved in those (P1)-(P13). What is nowadays not understood as being above OOP, can be characterired as the life rules and their switching; thus (P1)-(P2) must be understood in a reduced manner and (P3), (P8)-(P10) are to be ignored.

FURTHER STEPS ABOVE OOP

Already before 1967, Dahl and Nygaard realized the offer made by the dot notation and then overtaken by a lot of the OOP languages: (P11) and (P12) allow introducing new language statements without demanding to change the language compiler. Namely, *X.F(Y)* may represent a phrase where *X* is its subject, *F* its verb and *Y* its object or complement. In another context, the same expression can represent a clause where *F* is a copula connecting two related names etc. Dahl and Nygaard applied that offer e.g. to list processing, coming to statements like *A.precede(B)*, *A.follow(B)*, *A.into(S)* or A.out, and more evolved constructions like *R.first.suc.suc.follow(S.last.pred)*. Naturally, some later OOP languages use another character in place of dot – e.g. SmallTalk introduced a space.

From the practical point of view, the principles covered by common understanding to OOP are sufficient. But we have observed that it was already the "prehistory" of computer simulation that offered something that is above OOP – like life rules and their switching. In simulation of discrete and combined systems, the life rules are very fruitful. But they enabled including another principle; it was also discovered by Dahl and Nygaard an built into their language:

(P14) targets of transfers in performing the life rules can be virtual, i.e. occurring inside the life rules defined for a subclass.

If *C* is a class, (P14) enables the life of an object of its subclass *D* to "jump" among the life rules defined for *C*, for *D* and possibly for a subclass of *D* etc.; if *D* and *E* are two different subclasses of *C* then a target virtual in *C* may be interpreted differently for the instances of *D* and those of *E*.

Another principle accepted by the mentioned authors concerns the switching of life rules. All scheduling statements applicable in simulation can be defined as based upon a more general principle, which can be formulated as

(P15) introducing general standard procedures that enable any object *A* to interrupt performing its life rules and to switch the computer operation to another object *B*; later, when any object uses the same tool to switch the computer operation to object *A*, the life of *A* goes on from the place of its life rules where it left them the last time.

The users of popular tools for OOP like C++, SmallTalk and newer versions of Pascal battle with certain obstacles when they wish to describe something like life rules of objects or a complex dynamics of agents. In a better situation, the users of MODSIM III (Herring, 1990) are, which offers scheduling statements. Some OOP extensions of LISP (like C-Flavours) offer tools very similar to them mentioned in (P15).

BLOCK ORIENTATION AND OOP

Already the old SIMULA language followed the block structure existing in ALGOL 60 and the version of the new SIMULA did not neglect that attempt.

A class declaration exists often in the same context as the other declarations (those of variables and procedures). This phenomenon was followed in the new SIMULA until its complete seuqels and thus the folowing principles hold in this language:

(P16) class declarations can be local in the blocks, in which they occur; thus classes with the same name and with other semantics can occur in different blocks; it is especially appealing when one of these block is a subblock of the other;

(P17) declaration of a class *C* can occur in the same context to another class *D* as the other declarations occurring in class *D* (namely as the declarations of attributes and methods). The consequence is that every instance *X* of class *D* has its proper declaration of class *C* – similarly as it has its proper attributes; moreover, the declaration of *C* may depend on such attributes.

For speaking on principle (P17), the following terminology was introduced: *D* is called ***main class*** while *C* is characterized as class ***nested*** inside *D*. The instances of *D* represent entities that use *C* for thinking, modelling, communication etc. Usually *D* can have many nested classes and in such a case it represents a formal theory using the concepts represented by the nested classes.

Let among the life rules of class *E* be a block *B* with local classes. Then for any instance *G* of *E* the following phenomenon holds: when its life enters *B*, *G* behaves as being a specialist thinking by means of the local classes. When these classes use the tools for simulation then *G* can be considered as

(i) either a model of a computer which – having entered *B* – becomes a simulator using a model described in *B*,

(ii) or a model of a human which – having entered *B* – dynamically imagins some situation described and uses concepts mapped by the classes local in *B* for rational control of his imagining.

Let such an instance *G* exists contemporarily with other instances *G1, G2, ...*of *E* or of other classes declared at the same block (or object) *A* where *E* is declared, then *G* can be considered as an image of an element of a system modelled in *A* and composed of the mentioned instances mapped by *g1, g2, ...* Therefore such a manner is a way to implement "models in models" or "models containing modelling elements" (see Fig. 1).

All principles (P1) – (P17) have existed in the language defined in 1967. It was called SIMULA 67 and in 1986 –

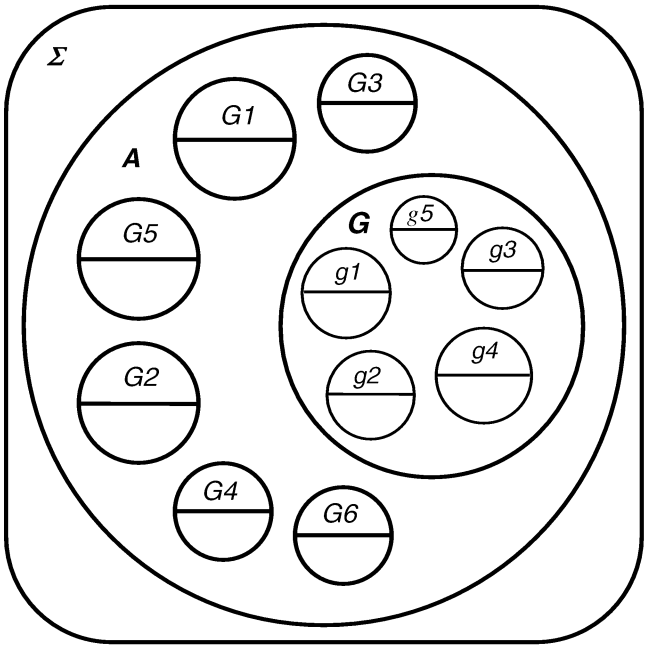


Figure 1. *Σ* is a block corresponding to a simulation study. Inside it, the object *A* represents a simulation experiment and (may be said) the external model of a certain system *S*, containing the representations of the elements *G*, *G1*,…,*G6*. At the "computer" *G* a model of *S* exists, too, but – in our example – the element represented by *G6* is neglected there.

when the language was registrated at ISO as an international standard and when the old simulation language SIMULA I was almost forgotten – the complement 67 was deleted. Similar principles can be observed at the language BETA (Madsen et al. 1993) and in JAVA. A disadvantage of BETAis its syntax which is very distant from customs and habits introduced in our civilisation by Fortran, Algol 60, C and other languages, disadvantages of JAVA consists in its rather poor OOP offers and in its direct absence of (P15), which leads to modeling this principle by means of real time programming. For modeling and simulation, a disadvantage of both BETA and JAVA is an offer to come into what happens in the computing system and dos not reflect anything what concerns the modelled system. Although such an offer can be beneficial in making software directly

related to the computer hardware (operation systems, computer networking etc.), it may introduce terrific wild run into (simulation) models.

APPLICATIONS

Almost every simulation application of SIMULA uses life rules and scheduling statements, i.e. some of its aspects that exceed the OOP. Essential applications of those aspects were simulation models of e.g.

(a) container yards with intelligently directed transporting tools (Kindler, 2000),
(b) circular conveyors existing in an intelligent environment (Kindler et al., 2004),
(c) hospitals using simulation to solve some "everyday" problems (Krivy and Kindler, 2006),
(d) queuing systems with simulating dispatchers (Kindler, 1996),
(e) personal transport in a region, reflecting the passengers' imagining in planning optimal traces (Bulava, 2002),
(f) automatic transport by intelligent induction carriages in production hall (Kindler and Brejcha, 1990),
(g) anthropologic metaphor of optimizing brain parameters (Faber and Weinberger, 1988),
(h) similar metaphor applied to optimization of production process in machine industry and in steel plants (Weinberger and Mojka, 1983).

ACKNOWLEDGEMENT

This work has been supported by the Grant Agency of Czech Republic, grant reference no. 201/060612, name "Computer Simulation of Anticipatory Systems" and by internal grant "Intelligent Information and Knowledge Systems and Their Implementation in Portals".

REFERENCES

Bulava, P. 2002. "Transport System in Havirov". In *Proceedings of 28th ASU Conference – The Simulation Languages*. FIT VUT, Brno CZ, 57-62

Buxton, J. N. (Ed.). 1968. *Simulation Programming Languages Proceedings of the IFIP working conference on simulation programming languages, Oslo, May 1967*. North-Holland, Amsterdam 1968

Dahl, O.-J. 1966. *Discrete Event Simulation Languages*. Norsk Regnesentralen, Oslo. Reprinted in (Genuys, 1968)

Dahl. O.-J. 2002. "The Birth of Object Orientation: the Simula Languages." In *Software Pioneers: Contribution to Software Engineering*, M. Broy and E. Denert (Eds.). Springer, Berlin, 78-90. Reprinted in (Owe et al, 2004), 15-25

Dahl, O.-J.; B. Myhrhaug; and K. Nygaard. 1968. *Common Base Language*, Norsk Regnesentralen, Oslo (1st ed.), 1972 (2nd ed.), 1982 (3rd ed.), 1984 (4th ed.)

Dahl, O.-J.; and K. Nygaard. 1965. *SIMULA – A Language for Programming and Description of Discrete Event Systems. Introduction and User's Manual*. Norsk Regnesetralen, Oslo.

Dahl, O.-J.; and K. Nygaard. 1966. "SIMULA – an ALGOL-based Simulation Language." *Communications of the ACM*, 9, No. 9, 671-678

Dahl, O.-J.; and K. Nygaard. 1968. "Class and subclass declarations." In (Buxton, 1968), 158-174

Faber, J.; and J. Weinberger. 1988. "Thalamocortical Reverberation Circuit Simulation Using the Simula Language." *Acta Universitatis Carolinae Medica*, 34, 149-248

Genuys, F. (Ed.). 1968. *Programming Languages*. Academic Press, London – New York

Gordon, G. 1961. "A General Purpose Simulation Program." In *Proc. 1961 EJCC*. MacMillan, New York, 81-98

Herring, C. 1990. "ModSim: A new object-oriented simulation language." *SCS Multiconference on Object-Oriented Simulation*. The Society for Computer Simulation, San Diego

Hoare, C. A. R. 1968. "Record Handling." In (Genuys, 1968), 291-346

Kalisz, E.; and A. M. Florea. 2000. "A GPSS simulation model of interactions in a market-based multi-agent system." In Urban, Ch. (Ed.): *Agent-Based Simulation. Proceeding of Workshop 2000, Passau, Germany, May 2000*. The Society for Computer Simulation International, San Diego, 2000, 145-150

Kindler, E. 1996. "Reflective Simulation (First Experiences)." In *Simulation und Animation für Planung, Bildung und Präsentation '96*. [ASIM Mitteilungen, Heft Nr. 54], ASIM, Magdeburg – Wien, 1996, 39-50

Kindler, E. 2000. "Nesting Simulation of a Container Terminal Operating With its own Simulation Model." *JORBEL (Belgian Journal of Operations Research, Statistics and Computer Sciences)*, 40, No. 3-4, 169-181

Kindler, E.; and M. Brejcha. 1990. "An application of main class nesting – Lee's algorithm." *SIMULA Newsletter*, 13, No. 3, (Nov.), 24-26

Kindler, E.; T. Coudert; and P. Berruet. 2004. "Component-Based Simulation for a Reconfiguration Study of Transitic Systems." *SIMULATION*, 80, 2004, No. 3, 153-163

Krivy, I.; and E. Kindler. 2006. "Synthesis of two Anticipatory Models in Design and Life-Cycle of Hospitals." *International Journal of Computing Anticipatory Systems*, 18, 75-85

Madsen, O. L.; B. Møller-Pedersen; and K. Nygaard, K. 1993. *Object-Oriented Programming in the Beta Programming Language*. Addison Wesley, Harlow – Reading – Menlo Park

Owe, O.; S. Krogdahl; and T. Lyche (Eds.) 2004. *From Obhect-Orientation to Formal Methods. Essays in Memory of Ole-Johan Dahl*. Lecture Notes in Computer Science, 2635, Springer, Berlin

Simula Standard. 1989. SIMULA a.s., Oslo

Weinberger, J.; and A. Mojka. 1983. "Optimization of an Industrial Simulation Model by Means of Quasi-Parallel Handling." *Ekonomicko-matematický obzor (Economical and Mathematical Review)*, 2, 179-185

EUGENE KINDLER studied at Charles University in Prague where he got a PhDr (doctor of philosophy) degree in logic and a RNDr (doctor of sciences) degree in mathematics. He worked as research specialist with Research Institute of Mathematical Machines in Prague (1958-66), and then as scientific specialist with Biophysical Institute at the Faculty of General Medicine of Charles University (1966-73) and with Faculty of Mathematics and Physics (1974-2000) of the same University. During 2000-2006, he was a professor of applied mathematics at the Faculty of Sciences of Ostrava University, with which he nowadays collaborates as emeritus. He was visiting professor with the University at Pisa (Italy), at Morgantown (West Virginia), at Clermont-Ferrand and at Lorient (France) and foreign lector of Humboldt University (Berlin). Czechoslovak Academy of Science gave him a degree CSc (Candidate of sciences) in physics and mathematics. He was the author of the first Czechoslovak ALGOL compiler and of the first Czechoslovak simulation system. Since 1967, his main interests have been oriented to the object-oriented simulation.

METHODOLOGY

SIMULATION METHODOLOGY

CRITERIA AND METHODS TO ESTABLISH THE VALID INTERACTION OF A SIMULATION AND ITS INTENDED PURPOSE

Vincent Albert, Alexandre Nketsa, Mario Paludetto and Marc Courvoisier
Laboratoire d'Analyse et d'Architecture des Systèmes LAAS-CNRS, Université de Toulouse
7 avenue du Colonel Roche
31077 Toulouse Cedex 4
France
E-mails: {valbert,alex,mario,myp}@laas.fr

KEYWORDS
M&S Hypotheses, Abstraction, Simplification, Validity, Experimental Frame

ABSTRACT

The main limitation in the evaluation of the validity and the composition/reuse of simulations is to formally describe what a model and the simulations constructed around this model represent. To solve this description problem, we have dedicated our investigations to the types of modelling and simulation hypotheses and their impacts on the simulation facilities. Indeed, a major thing which differentiates a simulation from the real world is the consequence of modelling and simulation hypotheses. The nature of these hypotheses defines which information was omitted and how it was omitted. The information omitted by the simulation defines its limitations. Vice versa, the information that the simulation conserves describes its usage domain.

INTRODUCTION

Context

The present study is inscribed in a wider project dedicated to assess the validity of using a simulation as a main Verification and Validation (V&V) process activity for an embedded system. Validity, as it is defined in most of the Verification, Validation and Accreditation (VV&A) studies (e.g. (DMSO, 2000)) refers to the degree that a model and its behaviour are suitable representations of the real system and its behaviour with respect to an intended purpose of model application (Brade, 2004).

The main expected benefits of the project are optimization of:

- The simulation use and development by designing the sufficient and necessary simulation product,
- The V&V process of systems by using the most relevant test means.

Such benefits imply on the one hand, to define relevant objects involved in the simulation process, and on the other hand to provide synthetic and accurate descriptions of user's intentions. Both have to be managed together in order to implement only relevant details.

These objectives will be achieved if a right methodology is defined to assist the development and the assessment of simulations. This methodology shall provide templates of simulation documentation and assessment techniques that will make simulation validation as automatic as possible. It shall assist the following steps:

1. Build the specification of the user's needs,
2. Identify and define the objects involved in the simulation process,
3. Define features, properties and constraints for each object,
4. Define criteria to validate, on the one hand the objects interaction and on the other hand the objects against user's requirements,
5. Check the former criteria by adequate methods.

The user's intentions (called in the following SOU "simulation objectives of use") have to be defined in terms of means, scenario (input, approximation, abstraction) and expected results.
The objects that are relevant for the project study can be: model, execution platform (EP) and the experimental frame (Zeigler, 2000) (EF).
The model documentation has to state about many factors such as : the type of the model (generic or executable), the atomic or coupled nature, hypothesis (with justification) about abstraction and/or approximation used, operating constraints, domain of use, the expected results, the scenario, the intrinsic properties as controllability, and observability. For example, a generic model means it is interpreted so the execution platform is soft, and an executable one means computer resources have to be taken into account therefore the execution platform must be checked for suitability.
Documentation of the execution platform can be a specification which describes the capacity of the platform in terms of operating system, computer power and communication resources.
Documentation of the experimental frame will deal with scenario that can be carried on, abstraction and/or approximation that can be made and expected results.

Then, criteria aim at establishing the valid interactions between tuple of objects or between object and user's needs. Examples of such valid interactions are consistency between model and EP, SEM and EF, EF and SOU (a SEM is a set of models implemented on an EP). We think it is possible to propose such connection right now (Figure 1).

Addressed issues

In this paper, we will address the problem of the valid interaction between a Simulation Executable Model (SEM) and an EF (the second criteria box showed in Figure 1). So, we assume that the EF is an adequate and relevant specification of a SOU, and that the EP does not alter the behaviour of the functional model.

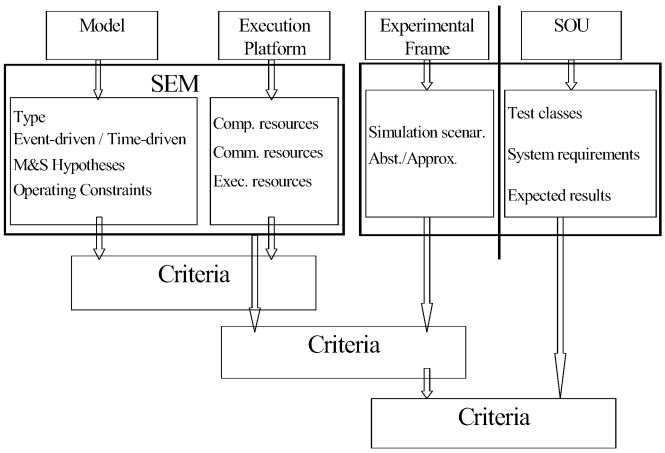


Figure 1 Global view of simulation products and their interaction

To study and validate the interaction between an EF and a SEM it is necessary that they have the same formalism of description. We specify this formalism of description through various dimensions, which each one, makes it possible to define a simulation at various levels of detail. For the moment, we have selected five dimensions taking from the Artificial Intelligence (IA) community (Weld, 1991): scope, depth, resolution, accuracy and temporal granularity. Some of these latter properties are also used in (DMSO, 2000) and (Roza et al., 2000). From the SEM point of view, these five dimensions represent its domain of use (SEM Domain of Use (SDU)). From the EF point of view, they represent the level of satisfaction of the objective of use of the simulation (what is needed, with which precision and which accuracy so that the fixed objective is satisfied?). So an SEM-EF interaction is valid, if the characteristics of the EF along these dimensions are consistent with the characteristics of the SEM along the same dimensions. This assessment requires establishing concrete criteria (Figure 2). It is worth noting this approach also allows managing the problem of the composition/re-use of models.

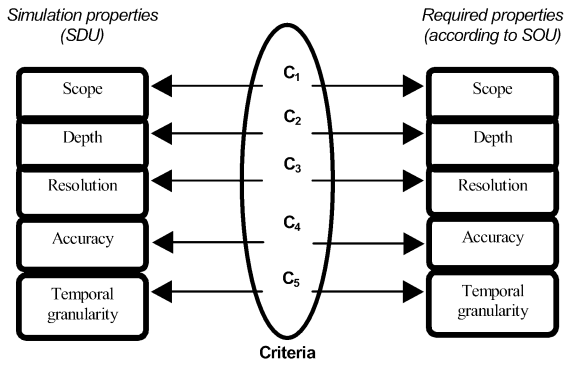


Figure 2 EF-SEM valid interaction

Such levels of details are increasing or decreasing according to modelling and simulation hypotheses. Then, in this paper, we propose a taxonomy of modelling and simulation hypotheses during the development of a simulation. Different taxonomies related to modelling have been previously proposed (Frantz, 1995)(Falkenhainer et al., 1991)(Harmon, 2006). Our taxonomy is built so as to ease the projection of the M&S hypotheses on the dimensions above.

In the first section, we introduce some preliminary definitions of the various terms used in this paper. The second section clearly defines the concept underlying the dimensions introduced above. The third section presents the types of modelling and simulation hypotheses which can affect these dimensions. These types of hypotheses are illustrated with concrete examples. The section four shows how we can, through the proposed dimensions and appropriate methods and metrics, assess the EF-SEM validity interaction.

The project study aims at identify main families of SOU (functional, performance/real-time, operational, interface/integration, safety…). Such a methodology described above must be applied for each class of SOU. Then, we must derive from such classes of SOU, an EF which will formally describe which are the expected results and the accuracy of these results. We are addressing in this paper only the functional class of test. It means that the characteristics defined in Figure 2, to express the expected results are requirements related to functional behaviour of the simulation only.

PRELIMINARY DEFINITIONS

Model

A model is a representation of selected aspects of the structure, behaviour, operation, or other characteristics of a real world process, concept, or system (Brade et al., 2004).

Component

A component is a basic element of a more complex unit, which is made of often different components. For example an aircraft fuel system is assemblies of different components such as pumps, tanks, quantity indicating system, refuel/defuel system.

Variables and relations

Conventionally, a mathematical or data processing model type of a physical phenomenon is specified by a set of algebraic equations (Travé-Massuyès et al., 1997). For example, a model for the electric circuit is specified by Ohm's law:

$$E = I.R$$

This equation includes a resistance R with a potential difference E at its terminals through which a current I passes.

Fundamentally, a model is therefore specified by a set of *relations* between the *variables* chosen to describe the system where each one of them must be defined over a range of values (e.g.: Real). Generally, a physical system is described by a set of variables the endogenous or exogenous character of which is determined by the chosen limits of the system in relation to its environment.

Static or dynamic model

Let us consider a physical system described by a set of exogenous variables and endogenous variables. The change

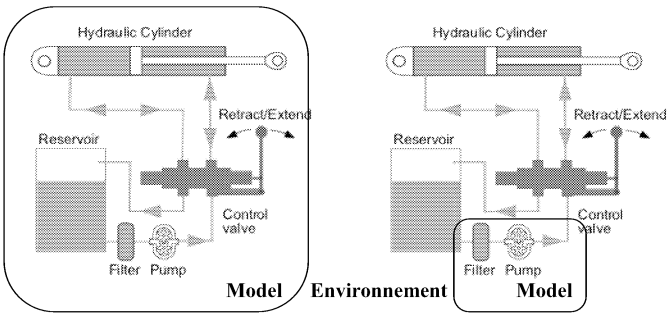
of its state, given by the values of the endogenous variables (where the minimum set allowing the system to be characterised represents the status variables), is conditioned by variations in the exogenous variables over time and by its own dynamics. With the stability hypothesis, the status changes from one equilibrium point to another, the systems own dynamics determining the type of change between two equilibrium statuses. A static model is a description of the system limited to its equilibrium states and is expressed in the form of functional relations between the status variables and the exogenous variables. A dynamic model describes the global change of the system including the transient statuses between two equilibrium statuses.
In many domains, the transients of the physical system can be neglected as the response times of the internal variables is sufficiently short in comparison with the variation rate of the exogenous variables (e.g. digital electronic circuit domain).

DEFINITIONS OF DIMENSIONS

Briefly, scope and depth refer to the functional chains and so on to the interfaces provided by the SEM. That consists in describing “what the model can do”. The resolution and accuracy refer to calculations. It consists in describing “how the model can do it”. Finally the temporal granularity refers to time, with the dynamics of the model.

Scope

By the scope of a model, we will denote the extent of the phenomena that the model describes. A model covers a larger perimeter than another if it describes more phenomena of the real world. For example, consider the hydraulic system model on Figure 3(a). It has a larger perimeter than the model of the electric pump and filter models on Figure 3(b). As the figure shows, changing the perimeter of the model is equivalent to distributing the limits between the system (described by the model) and the environment.



(a.) **(b.)**

Figure 3 Models with different scope

Depth

The depth relates to the level of details used in a model to describe a phenomenon. We will see later that we can do the abstraction of a model by aggregating some of its objects (components, relations, variables) depending on what we are trying to represent. When a collection of objects is aggregated in only one entity, then they become undistinguishable, as a consequence depth of the model decreases.

Resolution

The resolution reflects the level of detail used to express distinctions between the values of the variables describing the system. We can decrease the resolution relevant to a variable by reducing the cardinal of its value domain (for example, when we partition the set of real numbers by three signs: {- ,0,+}. Another typical example to illustrate the resolution is the number of digits after the decimal point that a datum can give.

Accuracy

The accuracy is related to the distance between the results of variables given by the model and the "reference" one, in the observed behaviour of the real system, for instance. Thus, we speak here about approximation and a more or less accurate model. For example, when a complex functional relation between internal variables of the model is replaced by a simpler one that give some less accurate values of some variables, this is called an approximation.

Temporal granularity

It makes it possible to evaluate if simulation represents time with a sufficient grain so as to capture the events of interests of the application. For example, one can choose not to represent the dynamics of certain variables, by considering that their response time is instantaneous compared to that of other variables. The corresponding derivatives are then eliminated from the model. The temporal granularity has been formally described by Ayrolles, Faivre and Guerrin (Ayrolles et al., 1996).

To illustrate four of these dimensions, let us take the simple example of a sensor making a level measurement in a tank the true value of which is H = 50 cm (Travé-Massuyès et al., 1997). In this trivial case, it is difficult to speak of scope and depth, the model being defined by a single variable H. If the model delivers value H = 50 cm, then it is accurate and has high resolution. However, if it gives a value of 48 cm, the model has still high resolution but it is inaccurate. Let us now suppose that the model gives the value of H in the form of an interval obtained by recording the lowest and highest level over the last 10 minutes and that this interval is equal to [46, 52], then the model has a lower resolution but it is accurate as the true value is in this interval. It is interesting to remark that we can restore the accuracy by decreasing the resolution.

MODELLING AND SIMULATION HYPOTHESES

Let us define two types of modelling and simulation hypotheses (Figure 4): abstraction and approximation. Abstractions and approximations are means of simplifying a system too complex in to order to adapt it to a given objective. The general idea is to class as abstraction the dimensions which reduce the observations of a model (scope and depth) and to consider as approximation all those affecting the calculations of the model (resolution, accuracy, temporal granularity).

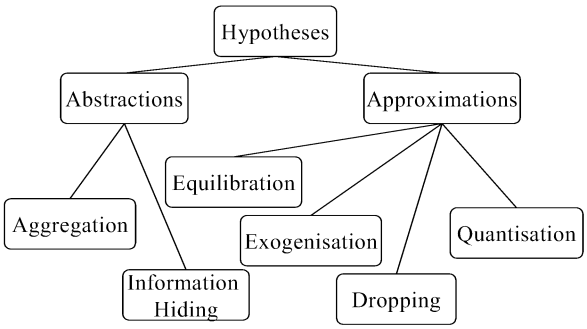


Figure 4 M&S hypotheses taxonomy

Abstractions

Abstraction is an essential technique in the modelling process to reduce a substantial amount of knowledge from the real world to serve a specific cause (e.g. simulation utilisation objectives). In general, it reduces the information available (depth) and increases the ambiguity but *without reducing* the accuracy. In other words, abstraction can imply some information loss but the remaining one is (and must be*) representative* of the real world; it is not the case with approximation. We have retained two main abstraction techniques: *information hiding* and *aggregation*.

Information hiding
Information hiding is an abstraction technique which reduces the complexity (in term of information quantity) of the model by hiding all objects that are uninteresting for certain investigations (Berard, 1993). Only the remaining objects are used for further analysis. The concept of hiding is widely used in Object-Oriented concepts. For example, in functional abstraction we might say that it is important to be able to add items to a list, but the details of how that is accomplished are not of interest and should be hidden. Using data abstraction, we would say that a list is a place where we can store information, but how the list is actually implemented (e.g., as an array or as a series of linked locations) is unimportant and should be hidden.

Consider for instance a system of systems (Figure 5). This system is composed of three sub-systems themselves composed of sub-systems. Assume that the objectives are to validate the system 2.1. This system is in interaction with the system 1.1, 1.3 and 3.1. Consider that Figure 6a illustrates an event graph for those systems at the lower level. According to the objectives, one can visualize completely only the system 2.1. On the other systems, only the events that establish connections with the system 2.1 are visualized. The remaining systems that do not communicate with the completely visualized system 2.1, are removed from the display and are therefore invisible (Figure 6b.) On Figure 5, that is traduced by considering the systems that interact with the system 2.1 at a higher level of details (middle level), retaining the events that establish connections with the system 2.1, but how the events are generated is unimportant and should be hidden.

Aggregation
Abstraction by aggregation consists in grouping various objects of a simulation into a single object, an aggregated entity. Abstraction by aggregation refers to the use of "high level" objects to represent a collection of "lower level" objects. In this type of relation, the higher level object has no grounds for existence in itself but is a representation of lower level objects. This type of abstraction produces new concepts by synthesising details into subconcepts.

All aggregation techniques are based on the same ideas:

1. We give ourselves a criterion according to which the objects can be considered as equivalent: in other words, we define an equivalence relation for all the objects.
2. We partition all the objects into equivalence classes.
3. For each class containing more than one element, we construct a more abstract object which groups all the characteristics of the objects of the class in question.

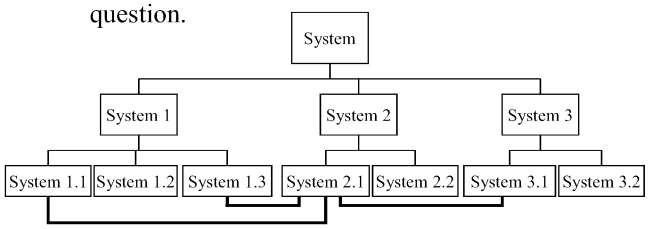


Figure 5 System hierarchical decomposition

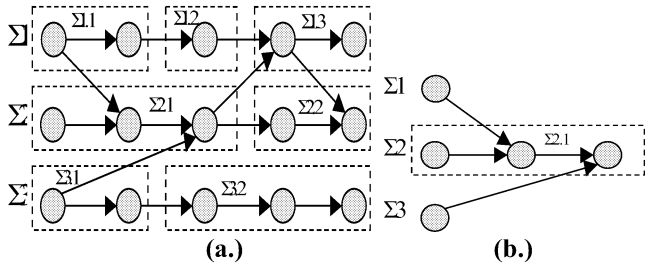


Figure 6 Models with different abstraction level

Let us consider, for instance, an aircraft fuel system which consists of several valves. Assume the objective of the model, is to supply the fuel flow rate consumed by the engines, to the component which calculates the weight of fuel in the aircraft. Thus, we can combine these valves into a single entity under the condition that the flow rate of this entity is associated with the sum of the flow rates of each valve. However, if this same model is used to determine from which tank an engine must take the fuel, this model will not be suitable as, in this case, and distinction must be made between the valves.

Approximations

Approximations are used to construct a simplified model typically simpler to use but to the risk of reduction of its resolution and/or accuracy. Generally, we associate approximation with computations (functions). In fact approximation consists in replacing a complex functional relation between internal variables of the model by a simpler but generally less accurate function. Indeed, an approximation can take several forms. Some consist in ignoring the influences which are (a priori) insignificant for a specific scenario. For example, calculating the flow rate from a fuel tank assumes that the pressure of the fuel is zero

and therefore ignores the effects of the fuel level in the tank on the outgoing flow rate. Another form of approximation can represent a simplifying hypothesis. For example, a tank can be modelled as an infinite source or a container of finite capacity.
The type of approximation used depends on the information available, the required degree of accuracy, the sensitivity of the problem to its data, the gain in time and effort which can be achieved. It is then a sort of trade-off between complexity and efficiency.

Dropping
Approximation by dropping consists in redefining the perimeter of the model by determining the minimum set of simulation objects (components, variables, statuses, relations), required for the simulation utilisation objectives. A model developer sometimes decides to neglect or ignore factors which seem to have sufficiently insignificant effects on the behaviour of the model for a specific utilisation objective and which would only complexify the modelling and simulation process.
Omitting objects may lead to the modification or the omission of others. Omitting a component implies the omission of all its descriptive variables. Omitting a variable implies the omission or at least the modification of the adjacent relations. If we omit a component or a variable which has no influence on an adjacent relation, we reevaluate the scope of the model. However, if this omission modifies an adjacent relation, there is an approximation and therefore the accuracy of the model must also be reevaluated.

Let us consider the block diagram shown on Figure 7 of an aircraft fuel system and the following simulation objective: study the movements of the aircraft and the flight handling qualities. The input data required for this study vis-à-vis the FUEL system are the inertia and the centre of gravity of each of the tanks. These data are measured and calculated by the FQMS component, Quantity Indicating and Fuel Storage Tanks. In this context of use, we can drop the other components and thus reduce the scope of the system to be modelled. Omitting some of these components induces the deletion of their variables. For example, the Fuel Storage Tanks component is influenced by the Fuel/Refuel component. Deleting it will generate "gaps" in all the relations (calculations) of the Fuel Storage Tanks component which used the variables produced by Fuel/Refuel. These gaps must be filled in one way or another.

Quantisation
An ordinary equation can be approximated as finely as possible by using quantisation. Quantisation is the process of approximating a continuous scale of values (or a very large set of possible discrete values) by a relatively small set of discrete symbols or integer values. The conversion of an analogue signal into a digital signal invariably induces errors. A continuous signal has an infinite number of value, with ideally an infinite accuracy, whereas the accuracy of a digital signal is dependent on the resolution of the quantisation.

Look-up tables are a specific discretisation case. Look-up tables are used to a great extent by model developers. A look-up table is a data structure, generally a table, used to replace a calculation (a relation) by a set of consultation operations. The input value is converted into an index of the table and the output value is determined from the indexed value. This type of abstraction enables a substantial gain in calculation speed. However, a table requires a fairly large quantity of memory space. The higher the number of points available, the higher the memory space required and the higher the accuracy.

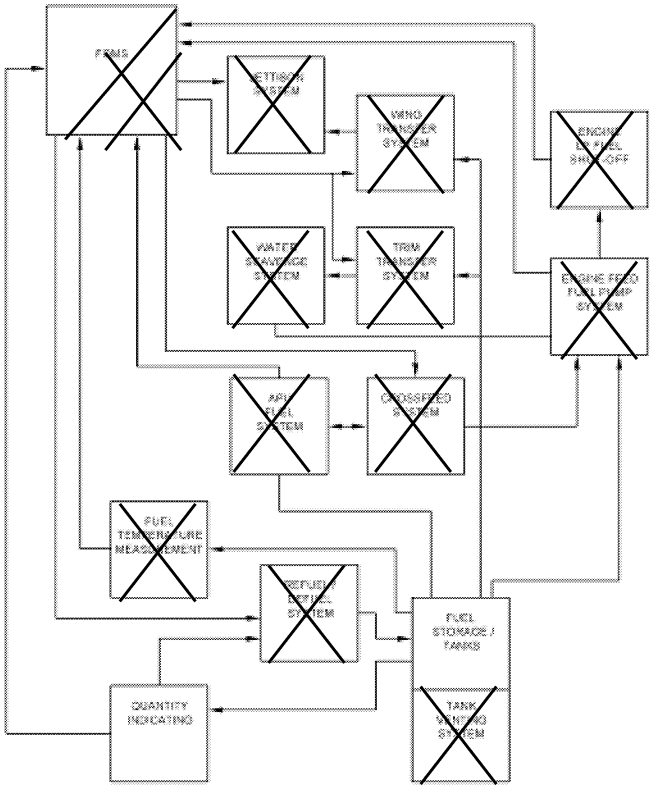


Figure 7 Approximated Fuel System Block Diagram

There are two basic limitations on when to use a table for a problem. The first one is the memory space available. The other restriction is the time required to initially calculate the values in the table. We can use less data points but the accuracy will be significantly reduced. A solution is to use interpolation, which consists in joining two points in the table and locates the result on the line drawn. This allows good accuracy to be conserved for the values which fall between two pre-calculated values for a computation speed as efficient as before.

Equilibration
Given a model, represented as a set of differential equations that describe the dynamic behaviour of a system, an approximation of that model, consists of some differential and some algebraic equations. When a mechanism represented by a differential equation in the dynamic model acts very quickly to restore relative equilibrium, one can regard it to be acting instantaneously when one is not interested in the very short-term dynamics of the system.

Let us take, for instance, a valve model for a system fuel. This model allows the opening and the closing of a valve to be controlled. This module gives two output variables: Valve_Open_Signal and Valve_Shut_Signal. The opening and closing of the valve require a certain Valve_Time.

When the valve is in open position, the Valve_Open_Signal is TRUE. When the valve is in closed position, the Valve_Shut_Signal is TRUE. At any other time (during Valve_time), Valve_Open_Signal and Valve_Shut_Signal are FALSE (Figure 7). There is an equilibration of the valve value position. The accuracy of the engine incoming flow rate calculation is of course affected by such an approximation as the valve is modelled as either Open or Shut whereas, in reality, it is of variable and therefore continuous resistance.

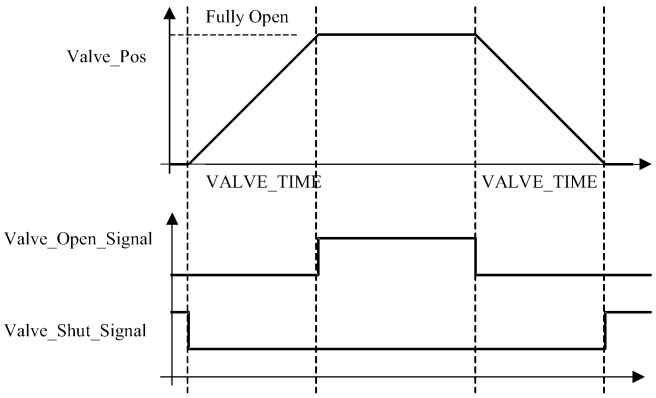


Figure 8 Valve position

Exogenisation

In contrast to variables that adjust to changes in other variables very quickly to restore relative equilibrium, some variables change so slowly in response to changes that they can be regarded as independent of other variables. The equation corresponding to such a variable can be replaced by a constant equation, which amounts to deleting from the system under consideration the slow mechanism through which others influence this variable. We call this operation of replacing a dynamic equation by an exogenous variable *equation exogenisation*.

Model developers used to use such approximation. For instance, fuel tank static head concept is not taken into account for the inter-tank flow rate calculation. The static head is the pressure exerted at output of a tank in relation to the fuel height. The variable corresponding to this pressure is approximated by a constant.

Equilibration and exogenisation are often called time-scale abstractions (Rickel et al., 1994)(Iwasaki, 1990). In both cases, the original system is simplified by replacing a differential equation by its corresponding equilibrium equation or by a constant. By doing these two operations, we can produce different models between the completely dynamic model and the completely static model according to the temporal granularity required for the objective of the analysis.

Table 1 recapitulates the types of hypotheses and dimensions which they would affect.

We have identified levels of details on five different dimensions which a SEM must be relevant for a given intended purpose and/or for a consistent interaction with other SEMs. The issue is now to provide relevant evidences to build the indisputable proof that the SEM degrees of details are relevant for the intended purpose. Assuming that the intended purpose (the user's need) has been formalised as required (EF), it consists in defining the sufficient and necessary conditions to ensure the valid interaction between EF and SEM.

Hypotheses / **Dimensions**	Abstraction		Approximation			
	Hiding	Aggregation	Dropping	Quantisation	Exogenisation	Equilibration
Scope	X	X	X			
Depth	X	X				
Resolution				X		
Accuracy			X	X	X	X
Temporal granularity					X	X

Table 1 Relation between hypotheses and dimensions

EF-SEM VALID INTERACTION ASSESSMENT

This section defines metrics and methods which can allow quantitatively or qualitatively assess the valid interaction between the EF and the SEM. The notion of EF-SEM duality has been addressed in (Traoré et al., 2005). The authors have formalised the EF in a DEVS-like fashion in order to ease the capture of the dual relationship between the EF and a simulation model described with DEVS.

Scope and depth

Define if a SEM has the required scope and the required depth with regard to an EF, are two issues which are bounded and which can be solved in the same way. We will use the concepts of controllability and observability that give very significant metrics for analysing depth and scope adequacy since they indicate what we can stimulate and observe with/on a given SEM.

Let us take the structure of a model M defined by the 4-tuple:

$$M = < E, V, R, S > \text{ where}$$

E is a set of input values
R: E x V $\rightarrow$ S x V' is the input/output relation
V is a set of observers on model states
S is a set of output values

Controllability: for a given V_0, there is always a sequence of E allowing S and V, to be obtained.
Observability: knowing S and V, we can find a given V_0.

In other words, a system is said to be controllable if we can act on all of its states by applying suitable stimuli. The observability tells us if all the states of the system are visible from existing observations.

R can be represented with state space form. A state space representation allows a dynamic system to be modelled in matrix form using *state variables* X1(t),…,Xn(t) which ensure the unicity of the causal relations E $\rightarrow$ S. The state

representation most generally used for linear systems is as follows:

$X'(t) = AX(t) + BU(t)$
$Y(t) = CX(t)$

where A is called the dynamic matrix, B the control matrix and C the observation matrix.

A system is said to be controllable if and only if the Kalman controllability matrix $K = [B \ AB \ \dots \ A^{n-1} B]$ has a non null determinant D(K).
A system is said to be observable if and only if the Kalman observability matrix $O = [C \ CA \ \dots \ CA^{n-1}]^T$ has a non null determinant D(O).

The applicability of an EF to a SEM corresponds to a relation of coupling between the specification of a SEM and the specification of an EF. By coupling, one obtains a new system of the form < E, V, R, S > made up of two subsystems (EF and SEM) as illustrated in Figure 9 (Traoré et al, 2005).

An EF refers to a set of stimuli and a set of observations. A stimuli-observation pair thus created must allow answering to a given question. It may be impossible to stimulate and/or observe phenomena and then reply to a question if an abstraction of the objects of the simulation eliminates the information required for the stimulation and/or the observation of the SEM. Also, we believe that the duality between EF and SEM can be analysed with controllability and observability metrics. If the system (consisting of the SEM and the EF) is controllable and observable, we believe that it is sufficient to say that the SEM scope and depth are consistent to be applied to a given EF.

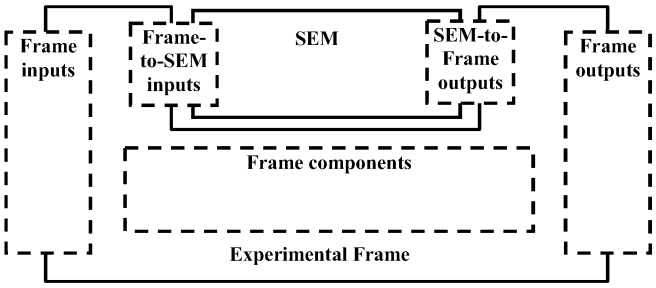


Figure 9 EF-SEM system

Resolution and accuracy

As the accuracy and the resolution are mainly based on calculations and therefore on mathematical (or data processing) functions, the evaluation of the accuracy and the resolution can be done by mathematical rules underlying the type of approximation used and the *errors* induced.

The approximation error of certain data is the difference between an accurate value and a certain approximate value or approximation of this. An approximation error can occur each time an approximation is made: the use of a look-up table is an example. If the input value does not exactly fall on one of the input values prerecorded in the table, an approximation is done and therefore an error is introduced. We believe than an approximation error on a variable can be traced and quantified. For example, the maximum approximation error produced by the use of a look-up table is the maximum difference between two output values of the table. A linear approximation is an approximation of a general function using a linear function (more precisely an affine function: the tangent). Mathematical methods allow the approximation error introduced by a linearisation to be evaluated (Taylor theorem).

We can then measure the *propagation of an approximation error* of a variable at the output of the model. The propagation of an error is the effect of an approximation error of a variable on the error of a relation (or of a calculation) based on this variable.

The Table 2 shows the uncertainty of simple functions, resulting from variables A, B, C with an error ΔA, ΔB, ΔC and a constant c.

$X = A \pm B$	$\Delta X = \Delta A \pm \Delta B$
$X = c.A$	$\Delta X = c.\Delta A$
$X = A.B$	$\frac{\Delta X}{X} = \frac{\Delta A}{A} + \frac{\Delta B}{B} + \frac{\Delta A.\Delta B}{A.B}$
$X = A.B.C$	$\frac{\Delta X}{X} = \frac{\Delta A}{A} + \frac{\Delta B}{B} + \frac{\Delta C}{C} + \frac{\Delta A.\Delta B}{A.B} + \frac{\Delta A.\Delta C}{A.C} + \frac{\Delta C.\Delta B}{C.B} + \frac{\Delta A.\Delta B.\Delta C}{A.B.C}$
$X = \ln(A)$	$\Delta X = \frac{\Delta A}{A}$
$X = e^{A}$	$\Delta X = e^{A}.\Delta A$

Table 2 Example of errors propagations

Temporal granularity

The sudden change from one temporal granularity level to another can mask intermediary representations which may be of interest. It appeared that there was a relation between the approximation operations of exogenisation, equilibration type and the causal order of a model (Iwasaki, 1990) and then on the accuracy of the simulation results.
The definition of causality, related to physical phenomena propagating in a structure, seems to agree on the three following criteria (Kleer, 1979):

- Temporality: the cause event must proceed the effect event,
- Locality: the cause acts only on its direct neighbours,
- Necessity: if the cause event has occurred, then necessarily the effect event must appear, all other things remaining equal elsewhere.

The equilibration and exogenisation definitions given above clearly illustrate the conditions under which we can use the corresponding approximation operation (e.g. exogenisation is used if the variable changes very slowly in response to a change). However, the problem is based on the order of magnitude of the "very slowly" notion. The work of Rickel and Porter (Rickel et al, 1994) proposes an automatic construction of models according to required prediction. From the specification of reference time scales for certain

variables, the system calculates the scales of the other variables which are coherent with the general model and allow a response to be given to the user's requests. It then determines, from among all the processes, those which will be considered as instantaneous and those which must be simulated dynamically.

In (Ayrolles et al, 1996), the authors have defined characteristics to give advice on choice of time scale: equilibrium, stability, periodicity, recurrence…

CONCLUSION

This paper has adressed the problem of assessing the adequacy of the domain of use of a simulation with regards to intended simulation objectives of use. We have proposed a general approach based on documentation template to assess such adequacy :

- SDU specification through a classification of modelling and simulation hypotheses and their related impact on simulation properties (scope, accuracy,…).
- SOU evaluation thanks to a classification of simulation objectives of use, i.e system properties to be explored through simulation (e.g. system logics, response time, …) + simulation properties required to serve the intended objectives of use.

We have proposed some criteria (e.g. controllability, observability) showing that effective properties related to SDU one by one match the required ones according to the SOU. Such properties and criteria must be refined and other validity indicators must be investigated. The interest of the proposed properties is that they can be affected on model interfaces. So, we would like to investigate on a plugin approach as it was initiated by Traoré (Traoré et al., 2005). This approach consists on build a simulation by interfacing models, execution platforms, and experimental frame. We think that an adequate documentation template of these simulation objects and the proposed methodology in this paper make it possible to define a priori the adequacy of a set of models on an execution platform with regards to intended simulation objectives of use.

REFERENCES

Ayrolles L., Faivre R., Guerrin F., 1996. “A temporal multi-resolution analysis for a quantitative/qualitative interpretation of complex dynamic process”. IEE Proceedings on Control Theory and Applications, 143 (2) : 191-199.

Berard, E., 1993. “Abstraction, Encapsulation, and Information Hiding”. in E. V. Berard, Object-Oriented Software Engineering. Englewood Cliffs: Prentice-Hall, 1993, 63-71.

Brade, D., 2004. “A Generalized Process for the Verification and Validation of Models and Simulation Results”. Dissertation, Faculty for Computer Science, University of the German Federal Armed Forces Munich, Neubiberg.

Brade, D. 2004. “VV&A Global Taxonomy (TAXO)”. Common Validation, Verification and Accreditation Framework for Simulation, REVVA.

Defense Modeling and Simulation Office (DMSO). 2000. “Verification, Validation and Accreditation (VV&A) Recommended Practices Guide (RPG).” Alexandria, VA. http://vva.dmso.mil/

Falkenhainer B., Forbus K. D., 1991. “Compositional modeling: finding the right model for the job”. Artificial Intelligence, v.51 n.1-3, p.95-143, Oct. 1991

Frantz F. K., 1995. “A taxonomy of model abstraction techniques”. Proceedings of the 27th conference on Winter simulation, p.1413-1420, December 03-06, 1995, Arlington, Virginia, United States

Harmon, S.Y., 2006. “Proposed Meta-Data for Describing Simulation Representational Capabilities”. Simulation Interoperability Workshops, 2006 Spring SIW, 06S-SIW-056

Iwasaki, Y., 1990. “Two Model Abstraction Techniques Based on Temporal Grain Size: Aggregation and Mixed Models”. Knowledge Systems Laboratory, August, 1990.

Iwasaki, Y., 1990. “On the relationship between Model Abstraction and Causality: Variance of Causal Ordering under Abstraction Operations”. Knowledge Systems Laboratory Report No. KSL 90-59, Stanford University, Sept. 1990

Rickel, J., Porter, B., 1994. “Automated Modeling for Answering Prediction Questions: Selecting the Time Scale and System Boundary”. Proceedings of the Twelfth National Conference on Artificial Intelligence.

Roza, Z. C., Gross, D. C., Harmon, S.Y., 2000. “Report Out of the Fidelity Experimentation Implementation Study Group”. 00S-SIW-151, *2000 Spring Simulation Interoperability Workshop (SIW)*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2000.

Traoré, M.K., Muzzy, A., 2005. “Capturing the dual relationship between simulation models and their context”. Simulation modelling practice and theory.

Travé-Massuyès, L., Dague, P., Guerrin, F., 1997. “Le raisonnement qualitatif pour les sciences de l’ingénieur”. Broché,05-1997, ISBN : 2-86601-611-4

Weld, D. S. 1991. “Reasoning about model accuracy”. TR 91-05-02, University of Washington, Department of Computer Science and Electrical Engineering

Zeigler, B.P., H. Praehofer, and T.G. Kim. 2000.“Theory of Modeling and Simulation”. Second Edition. Academic Press. ISBN: 0-12-778455-1

APPLICATION OF SYSTEM DYNAMICS METHODOLOGY FOR DETERMINATION OF SUSTAINABLE SARDINE CATCH IN EASTERN ADRIATIC

Merica Sliskovic
Ante Munitic
Gorana Jelic-Mrcelic
Faculty of Maritime Studies Split, University of Split
Zrinsko-Frankopanska 38, 21000 Split, Croatia
E-mail: merica@pfst.hr

KEYWORDS

Natural resources, Decision-making, Dynamic modelling
Combined simulation, System dynamics

ABSTRACT

Qualitative and quantitative models of Adriatic Sea sardine population are made by using biological data available from literature. Schaefer production model for sardine population is made by the System Dynamics methodology. Several simulation scenarios are made in Powersim and DYNAMO program package, in order to determinate optimal sardine catch. Models developed in this paper shows that high catchability coefficient values do not provoke high catch, but low values of catchability coefficient cause high values of biomass. It is strongly recommended not to assume catchability coefficient as a constant but to take into account that catchability increases constantly every year.

INTRODUCTION

Modeling and simulation enable testing of different exploitation scenarios without endangering fish population in reality. The Schaefer production model (1954) is often applied in order to submit behavior dynamics of fish population and particularly to manage marine natural resources efficiently. Alegria-Hernandez (1983) also used Schaefer and Fox production models for sardine population in Eastern Adriatic. In this paper, Schaefer production model is used due to the lack of appropriate biological data for any other model.

Production models are very simple but they can be good approximation for complex behavior dynamics of biological systems. Change of population biomass (Dudley, 2003), which mathematically matches first derivation, is:

$$\frac{dB}{dt} = \left[(r \cdot B) - (r \cdot \frac{B^2}{CC}) \right] - C \qquad (1)$$

where dB/dt is the rate of biomass change, r intrinsic growth rate, CC carrying capacity, C catch rate and B sardine biomass. Catch rate is defined as product of sardine biomass, catchabilty coefficient and fishing effort (Ussif, 2003). Initial values for sardine biomass, carrying capacity, catchabilty coefficient and fishing effort are given according to Alegria-Hernandez (1983). Catchability coefficient (q) represents the probability that the fish will be caught and is defined as relation between catch per unit effort (CPUE) and size of population (B). It is also known as the efficiency of fishing gear (Hilborn et al., 1992). Santojanni *et al.* (2001) use two different values of catchability coefficient obtained by DeLury method (2.17E-5 and 3.51E-5) for the estimation of Adriatic sardine population and those two values are used for testing the influence of catchability coefficient on catch value in this paper.

The structural model of sardine population that presents all variables included in the model, together with existing causes-consequences links and feedback loops in sardine population system is given in Figure 1.

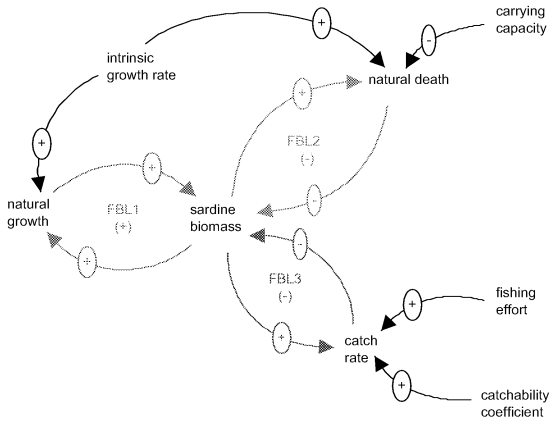


Figure 1: System Dynamics Structural Model of Eastern Adriatic Sardine Population

System dynamics structural flow diagram of sardine population is presented in Powersim language is given in Figure 2.

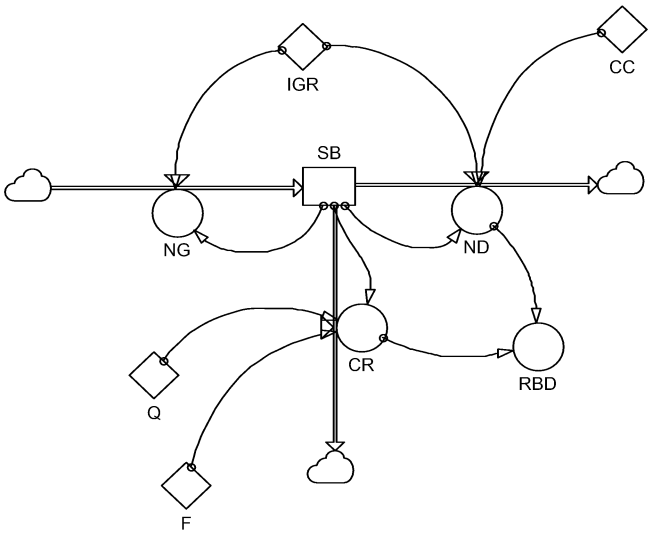


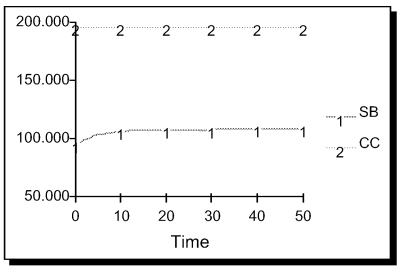
Figure 2: System Dynamics Flow Diagram of Sardine Population in Powersim Language

Where symbols present NG natural growth, SB sardine biomass, IGR intrinsic growth rate, CC carrying capacity, ND natural death, CR catch rate, Q catchability coefficient, F fishing effort, RBD rate of biomass decrease.

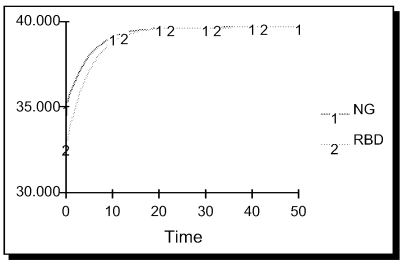
RESULTS

Figure 3 shows behavior dynamics of sardine system when catch is product of catchability coefficient q=3.99E-5 and fishing effort f=4115 effective fishing days in one year - **Scenario 1**.

a)



b)



c)

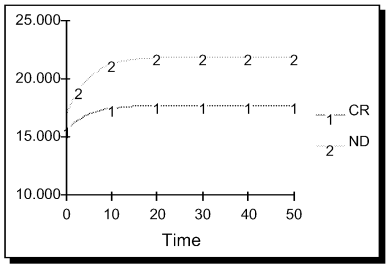
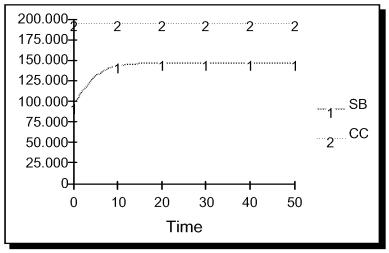


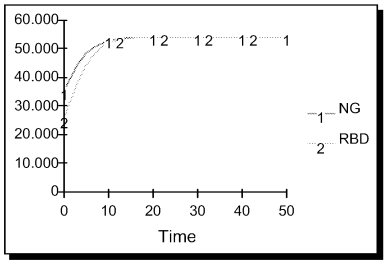
Figure 3: Results of Scenario 1 for: a) sardine biomass (SB) and carrying capacity (CC); b) natural growth (NG) and rate of biomass decrease (RBD); c) catch rate (CR) and natural death (ND)

Figure 4 shows behavior dynamics of sardine system when catchability coefficient is q=2.17E-5 and fishing effort is f=4115 effective fishing days in one year (**Scenario 2**).

a)



b)



c)

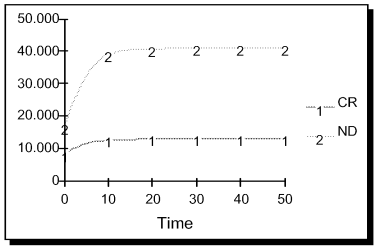
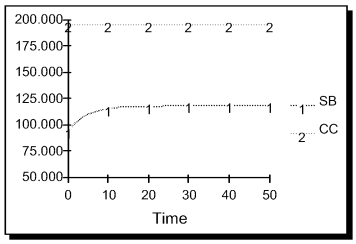


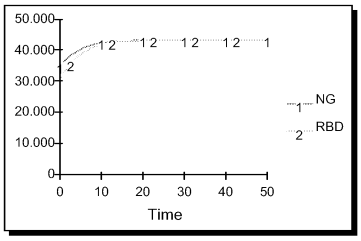
Figure 4: Results of Scenario 2 for: a) sardine biomass (SB) and carrying capacity (CC); b) natural growth (NG) and rate of biomass decrease (RBD); c) Results of Scenario 2 for catch rate (CR) and natural death (ND)

In **Scenario 3** (Figure 5), behavior dynamics of sardine system is investigated when catch is product of catchability coefficient q=3.51E-5 and fishing effort f=4115 effective fishing days in one year.

a)



b)



c)

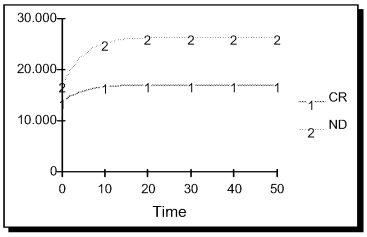


Figure 5: Results of Scenario 3 for: a) sardine biomass (SB) and carrying capacity (CC); b) natural growth (NG) and rate of biomass decrease (RBD); c) catch rate (CR) and natural death (ND)

DISCUSSION

Scenario 2 (Figure 4a) shows that value of sardine biomass increases under fishing intensity, i. e. catch as a product of q=2.17E-5 and f=4115, at the beginning of the simulation and after some time reaches steady state lower than in Scenario 1 (Figure 3a). In the beginning of the simulation positive feedback loop (FBL1 (+)) dominates and cause increasing of biomass, i.e. natural growth is higher than rate of biomass decrease. These two values equals as system reach steady sate (Figure 4b). During the simulation natural death ND and catch rate CR increase and after short period ND amounts almost twice of the initial value, while CR amount 1.5 of the initial value. If and natural death are compared (Figure 4c) than natural death dominates over the catch. The value of catch rate is almost twice smaller than the value of natural death. The behavior dynamics in the Scenario 3 corresponds to those in Scenario 2 but initial sardine biomass increase differs because the biomass reaches steady state at lower value. Comparative analysis of the natural death and catch rate (Figure 5c) shows that natural death is higher than catch rate, but there is less difference compared to Scenario 2, probably due to the increase of catch rate. Higher values of catchability coefficient do not cause increase of the catch (Figure 4c and 5c), but lower values of catchability coefficient causes higher values of biomass (Figure 4a). Major assumption in the use of models is that the relationship between catch rates and stock biomass implies that the catchability coefficient is a constant, but it is found a constant proportional increase in catchability each year in some fisheries, because of continual improvements to fishing gear and fishing practices (effectiveness of each unit of effort) (Haddon, 2001). Therefore, it is strongly recommended not to assume the catchability coefficient as a constant.

CONCLUSION

Although very simple, Schaefer production model enables investigation of complex behavior dynamics of fish population. Qualitative and quantitative models developed in this paper show that lower values of catchability coefficient cause higher values of biomass, while higher values of catchability coefficient do not have influence on the value of catch. It is strongly recommended not to assume the catchability coefficient as a constant. It is also proved that by system dynamics modeling different values of the catchability coefficient can be tested. Thus the critical value of catchability coefficient (optimal catch quotas) can be determined and sustainable development of observed population achieved without endangering real population.

REFERENCES

Alegria-Hernandez, V. 1983. “Assessment of pelagic fish abundance along the eastern Adriatic Sea cost with special regard to sardine (Sardina pilchardus, Walb.) population”, Acta Adriatica, No.4, 55-95.

Dudley, R. G. 2003. “A Basis for Understanding Fishery Management Complexities” (27.10.2005) [http://pws.prserv.net/RGDudley/PDF/FMTcompl.pdf]

Haddon, M. 2001. “Modeling and Quantitative Methods in Fisheries” Chapman & Hall, London.

Hilborn, R. and C. J. Walters. 1992. “Quantitative Fisheries Stock Assessment: Choice, Dynamics, and Uncertainty” Chapman & Hall, London.

Santojanni, A., E Arneri, A.Belardinelli, N.Cingolani and G. Giannetti. 2001. “Fishery and stock assesment of *Sardina pilchardus* (Walb) in the Adriatic Sea “. Acta Adriatica, No.42 (1), 151-168.

Schaefer, M. B. 1954. “Some Aspects of the Dynamics of Populations important to the Management of Commercial Marine Fisheries”. Bulletin of the Inter - American tropical tuna commission, No. 1, 25-56.

Ussif, A.A., L. K. Sandal and S. I. Steinshamn. 2003. “A new approach of fitting biomass dynamics models to data”. Mathematical Biosciences, No.182 (1), 1-25.

MODELLING DISCRETE INPUT PROBABILITY DISTRIBUTIONS

Johann Christoph Strelen
Rheinische Friedrich–Wilhelms–Universität Bonn
Römerstr. 164,
53117 Bonn, Germany
strelen@cs.uni-bonn.de

KEYWORDS

Stochastic Simulation, Fitting Probability Distributions, Discrete Probability Distributions, Genetic Algorithm, Inverse Transformation Method.

ABSTRACT

Sometimes input probability distributions for stochastic models are not so simple that standard distributions suit. In this case, we model with weighted sums of standard distributions. These composed distributions may have many parameters which must be estimated. This is not easy with common estimation methods like maximum-likelihood. We use the genetic algorithm for that. The design of the composed distributions can be done such that the inverse tranformation method for the generation of random variates is feasible.

1 INTRODUCTION

Stochastic models and discrete simulation are indispensable means for the quantitative analysis of systems. It is well known that missing to carefully model the influences from outside, especially the load, may lead to wrong results and ultimately to wrong decisions based on the simulation results.

Influence from outside of the model like load or failure of system components can be incorporated into the model using observed traces or input models, namely independent random variables, random vectors, or stochastic processes. Data from traces can be used directly.

If input is modeled, input data are realisations of the model. The use of random variates which are drawn from common standard distributions, is well understood and common since long time, see any book on discrete simulation like (Law and Kelton 2000). The use of generated random vectors and stochastic processes is much more difficult, not so popular, a topic of current research, see for example (Strelen and F.Nassaj 2007) and the citations there.

In this paper, we consider independent random variates with more general distributions which are composed of some standard distributions, with emphasis on discrete distributions. We show that such composed distributions are suitable for the inverse transformation method for generating random variates.

The common way to select a standard distribution is as follows. A sample of data is gathered in some real system, and a standard distribution is searched from which the data could be realizations. Usually this is done in three steps: In the first step, a specific standard distribution is hypothesized, for example Weibull with two parameters. In the second step, the distribution is fitted to the data, namely distribution parameters are estimated. In the third step, one must decide if the found distribution is representative for the data, for example with goodness of fit tests.

The ExtertFit software (Law and Kelton 2000) determines which of 39 considered probability distributions represent best some given data. To this end, the distributions are fitted to the data using maximum-likelihood estimators, and the quality of each fit is judged with goodness of fit tests.

But it may happen that there is no suitable standard distribution for the observed data. The data may indicate more than one maximum of the density or the probability function, but all standard distributions are unimodal, hence none of them fits. Or, a discrete sample is very irregular, some values occur often, the others in between seldom.

Under these circumstances, a composed distribution, namely a weighted sum of standard distributions, may be appropriate. Such a weighted sum is convenient for generating random variates.

Moreover, a distribution which is composed of standard distributions with invertible distribution functions may be invertible itself. In this case, the inverse-transform method can be applied to generating random variates. This allows for some variance reduction methods.

But such a composed distribution has many parameters which must be estimated, in general. This is a difficult task, since the common maximum-likelihood estimators lead to a system of nonlinear equations which must be solved; the number of equations equals the number of unknown parameters.

In (Strelen 2003), we proposed to apply a genetic algorithm instead which has advantages compared to the tradi-

tional way. We develop this idea further in this paper. Here we only need to distinguish three slightly different techniques, one for closed-form distribution functions, the second with numerical integration of densities, and the third for discrete probabilities - on the other hand, maximum-likelihood estimators require many individual solutions for different distributions. And, even more important, many parameters can be considered easily with the new technique.

In the first place, the genetic algorithm serves the purpose to estimate parameters in our technique, but as a byproduct, standard distributions can be judged if they are suitable, and one obtaines a measure how good the fit is.

The genetic algorithm (Back, Fogel, and Michalewicz 1997, Davis 1991, Goldberg 1989) is a stochastic global search method that mimics the metaphor of natural biological evolution. We use it to minimize the distance between an empirical distribution of the observed sample and the (composed) distribution which is to be fitted.

For numerical calculation we use MATLAB (Matlab) with the Genetic Algorithm Toolbox (Chipperfield, Fleming, Pohlheim, and Fonseca).

In section 2, we report on the genetic algorithm as far as important for our method. Section 3 describes the fitting technique, gives some examples, and points out the applicability of the inverse transformation method. Section 4 is up to an intricate example: We consider a measured sample of packet sizes in a computer network from Klemm, Lindemann, and Lohmann (2002).

2 THE GENETIC ALGORITHM

The genetic algorithm (GA) is a stochastic global search method that mimics the metaphor of natural biological evolution. It differs from traditional search and optimization methods, significant differences are:

- Genetic algorithms search generations of approximations in parallel, not a single sequence
- Genetic algorithms require only the objective function, no derivatives
- Genetic algorithms use probabilistic transition rules, not deterministic ones
- Genetic algorithms work on an encoding of the parameter set rather than the parameter set itself

Genetic algorithms (Back, Fogel, and Michalewicz 1997, Davis 1991, Goldberg 1989) operate on *populations* of individuals applying the principle of survival of the fittest. *Individuals* are tuples of decision variable values which are encoded as strings over the binary alphabet or other alphabets. The individuals are approximations of the desired solution, and their *fitness* measures the acuracy which is fixed by an objective function. At each *generation*, a new population is created by the process of selecting individuals according to their level of fitness and breeding them together using operators borrowed from natural genetics. The *recombination operator* is used to exchange parts of the strings between pairs, or larger groups, of individuals, according to some probabilistic rule. *Mutation* will cause a single bit to change its state with some probability, in the binary string representation. *Selection* serves the purpose to select individuals for the next generation.

This process leads hopefully to the evolution of populations of individuals that are better approximations to the solution than their parents.

We use the Genetic Algorithm Toolbox (Chipperfield, Fleming, Pohlheim, and Fonseca) which, broadly speaking, works as follows. In each problem which is to be solved here, the genetic algorithm is applied to searching for a minimum of the objective function. The result is a tuple of values of the real valued decision variables d_i, $i = 1,\ldots,N^{(\mathrm{var})}$. For each decision variable, a minimum and a maximum is given in advance, and the values are encoded with *PRECI* binary digits. Grey coding is used, hence adjacent values differ in just one digit. An $N^{(\mathrm{var})}$-tuple of decision values is an individual, and $N^{(\mathrm{ind})}$ individuals are a population. The encoded values of an individual are concatenated, hence form a binary string of $N^{(\mathrm{var})} * PRECI$ digits which is termed the chromosome of the individual. The chromosomes of all individuals of a population are the matrix *CHROM* with $N^{(\mathrm{ind})}$ rows, each a chromosom.

The genetic algorithm works as follows. The matrix *CHROM* is initialized with uniformly distributed random numbers. For each individual, its chromosome is decoded into a decision variable tuple $\mathbf{d}$, and the objective function $Z(\mathbf{d})$ is calculated.

In a loop, *MAXGEN* populations are calculated, one after the other: For each individual, the fitness with linear ranking is calculated according to the value of the objective function $Z(\mathbf{d})$ of its decision variables, as follows. The individuals are sorted according to descending values of their objective function values, i.e. the best individual gets the position $pos = N^{(\mathrm{ind})}$. Each individual on position *pos* gets the rank $2(pos-1)/(N^{(\mathrm{ind})}-1)$, rank 2 is the best, rank 0 the worst.

GGAP, $0 < GGAP \leq 1$, is the generation gap: The $(1 - GGAP) * N^{(\mathrm{ind})}$ individuals with the best fitness values remain unchanged, and the other $GGAP * N^{(\mathrm{ind})}$ individuals are selected for breeding offspring with the method of stochastic universal sampling: Each individual i gets a probability p_i which is proportional to its fitness value. The selection is according to these probabilities where individual i is selected at least $\lfloor p_i N^{(\mathrm{ind})} GGAP \rfloor$ times and at most $\lceil p_i N^{(\mathrm{ind})} GGAP \rceil$ times ($\lfloor . \rfloor$ means the floor, and $\lceil . \rceil$ the ceil).

The selected chromosomes are pairwise recombined with the single-point crossover operator, each pair with probability 0.7, and with probability 0.3, the two individuals of a pair remain unchanged with respect to recombination. If the number of selected individuals is odd, one more remains unchanged.

For this crossover, a position within the two chromosomes of a pair is selected at random, and the left part of one chro-

mosome is concatenated with the right part of the other, and vice versa. The result are two new chromosomes.

Now on all chromosomes, the mutation operator is applied. Each binary digit flips with probability $0.7/L^{(\mathrm{ind})}$ where $L^{(\mathrm{ind})} = N^{(\mathrm{var})} * PRECI$ is the number of binary digits in each chromosome.

The objective function values are now calculated for offspring.

The reinsertion step performs insertion of offspring into the current population, replacing least fit parents with offspring.

Now the next population is ready and the process continues if the number of populations is less than $MAXGEN$.

The individual with the best objective function value is the result, and this value measures the accuracy of the fitted probability distribution.

3 FITTING INPUT PROBABILITY DISTRIBUTIONS

In principle, we proceed as usual but with two main differences: Our fitted distributions are more complex and, in general, have many parameters which must be estimated, and we search with the genetic algorithm.

As basis, a sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of data is given which was measured for a specific aspect of a real system. Then a composed distribution is selected and fitted to the data,

$$\begin{aligned} F(x) &= q_1 F_1(x) + q_2 F_2(x) + \ldots + q_K F_K(x), \\ & \quad q_k > 0,\ q_1 + q_2 + \ldots + q_K = 1, \end{aligned} \tag{1}$$

where F_1, $F_2, \ldots$ are standard distribution functions (CDF).

But it may be sensible to modify the standard distributions. A location parameter β or a scale paremeter γ may be introduced if not present. Instead of the random variable X with the CDF $F_k(x)$ then we consider $Y = \gamma X + \beta$ with the CDF $F_k((x-\beta)/\gamma)$ and the density $f_k((x-\beta)/\gamma)/\gamma$ where it exists. Moreover, the distribution may be cut, we consider the random variable Z with the CDF

$$F_Z(x) = \begin{cases} F_k(x)/F_k(\eta), & x \le \eta, \\ 1, & \eta < x. \end{cases}$$

β, γ, and η are additional parameters.

For the fitting, the parameters q_k and parameters for each standard distribution F_k are estimated. We denote the tuple of these decision variables with $\mathbf{d}$ and indicate the dependency of the distribution function sometimes writing $F_{\mathbf{d}}(x)$.

For discrete random variables with integer values, the composed distribution is

$$\begin{aligned} p(x) &= q_1 p_1(x) + q_2 p_2(x) + \ldots + q_K p_K(x), \\ & \quad q_1 + q_2 + \ldots + q_K = 1, \\ & \quad x = \ldots, -1, 0, 1, \ldots, \end{aligned} \tag{2}$$

where $p_1(x)$, $p_2(x), \ldots$ are (modified) standard distributions.

Once the decision variables are estimated one decides if the fitted distribution (1) is correct.

A random number with such a composed distribution can be generated as follows: First draw a random number k with the discrete distribution $(q_1, \ldots, q_K)$. Secondly, draw a random number with the distribution $F_k(x)$. We will say something more about generation of random variates in the subsection.

For fitting with the GA, we need an empirical distribution. We use a specific kind of empirical distribution functions, namely step functions. They are defined with the order statistics as follows:

$$\hat{F}(x) = \begin{cases} 0, & x < x_{(1)}, \\ i/n, & x_{(i)} \le x < x_{(i+1)},\ i = 1, \ldots, n-1, \\ 1, & x_{(n)} \le x. \end{cases} \tag{3}$$

Here, $x_{(i)}$, $i = 1, \ldots, n$, denotes the ordered sequence with the elements x_i.

Remark 1. If in the ordered sequence $x_{(i)} = x_{(i+1)}$, then there is no x for which $\hat{F}(x) = i/n$, i/n is not in the range of $\hat{F}$. This occurs due to finite accuracy of real numbers in the computer and with discrete samples.

Remark 2. If all sample values $x_{(l)}$, $l = 1, \ldots, n$, are different, $\hat{F}(x_{(l)}) = l/n$ holds. If not, this is true only for values $x_{(l)}$ which occur only once, and for just one of some equal values. Namely for multiple values $x_{(l_1)} = x_{(l_2)} = \ldots$, $\hat{F}(x_{(l_i)}) = \max\{l_j/n\}$ holds for $i = 1, 2, \ldots$.

For discrete random variables with integer values, we use frequency distributions,

$$\hat{p}(x) = \hat{F}(x) - \hat{F}(x-1),\ x = \ldots, -1, 0, 1, \ldots. \tag{4}$$

Again we use the notation $p_{\mathbf{d}}(x)$ to indicate the dependency of the parameters.

For continuous distributions, our objective function for the genetic algorithm is

$$Z(\mathbf{d}) = \sum_{i=1}^{n} [\hat{F}(x_i) - F_{\mathbf{d}}(x_i)]^2 \tag{5}$$

where $F_{\mathbf{d}}(x)$ is the selected (composed) distribution function with the parameter tuple $\mathbf{d}$.

$\sqrt{Z(\mathbf{d})/n}$ is a L2-norm of the difference between the empirical distribution function and the fitted distribution function and measures how accurate the distribution function $F_{\mathbf{d}}(x)$ fits the given data.

Sometimes there is no closed form of a standard distribution function available, for example if it is Gamma or Lognormal. Here we use the density $f_k(x)$ and calculate the distribution function $F_k(x)$ at the values x_i, $i = 1, \ldots, n$, approximately with very simple numerical integration. In particular, we take

$$\begin{aligned} \tilde{F}(x_1) &= 1/n, \\ \tilde{F}(x_i) &= 1/n + \sum_{j=2}^{i} (x_j - x_{j-1}) f_k(x_{j-1}),\ i = 2, \ldots, n, \\ F_k(x_i) &\approx \tilde{F}(x_i)/\tilde{F}(x_n),\ i = 1, \ldots, n. \end{aligned}$$

For discrete samples, the objective function is

$$Z(\mathbf{d}) = \sum_{i=1}^{n} [\hat{p}(x_i) - p_{\mathbf{d}}(x_i)]^2. \tag{6}$$

In the GA algorithm, for each decision variable, the range of possible values must be given in advance. Sometimes this is not obvious. The weights q_k in (1) are restricted to $q_1 + q_2 + \ldots + q_K = 1$ - what is the range of q_k? We consider the parameters u_k instead which have the unit range, $0 \le u_k \le 1$:

$$\begin{aligned} q_1 &= u_1, \\ q_2 &= (1-u_1)u_2, \\ q_3 &= (1-u_1)(1-u_2)u_3, ..., \\ q_K &= (1-u_1)(1-u_2)...(1-u_{K-1}). \end{aligned}$$

If there are parameters a_1 and a_2 with $0 \le a_1 \le a_2$, we replace a_1 with $b_1 a_2$, $0 \le b_1 \le 1$. Moreover, we replace an integer valued parameter j with $j = \lfloor y \rfloor$, y real valued.

For the numerical calculation we used MATLAB (matlab) with the Genetic Algorithm Toolbox (Chipperfield, Fleming, Pohlheim, and Fonseca) which provides very comfortable functions for genetic algorithms. It remains mainly to specify the number of decision variables (parameters), their type (integer or real), their degrees of accuracy (number of binary digits), and an objective function. Further specifications concern the features of the genetic algorithm like the number of generations, the number of individuals in a population, and others.

Example 1, Fitting Data Drawn from a Two-Mode Weibull Distribution. The sample consists in 800 realizations according to the mixed distribution function $F(x) = 1 - 0.5\exp[-(x/3)^2] - 0.5\exp[-(x/17)^5]$, the fitted distribution function (figure 1) is $F(x) = 1 - 0.496\exp[-(x/3.03)^{1.90}] - 0.504\exp[-(x/17.1)^{5.46}]$. The accuracy is 0.004, see also figute 1.

In figures 1 and 3, the smooth curve is the fitted theoretical distribution function, and the scribbling curve which is shifted by 1 to the right in order to separate the graphs is the empirical distribution function.

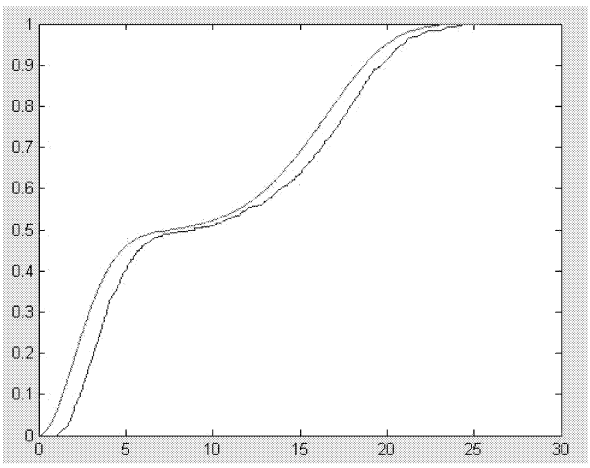


Figure 1: A Fitted Two-Mode Weibull Distribution Function

Example 2, Fitting Data drawn from a Three-Mode Poisson distribution The sample consists in 64000 realizations from the distribution

$$p(x) = 0.333p_8(x) + 0.333p_8(x-16) + 0.334p_8(x-32),$$

where

$$p_\lambda(x) = \begin{cases} 0, & x < 0, \\ e^{-\lambda}\lambda^x/x!, & 0 \le x, \end{cases}$$

the Poisson distribution. 16 and 32 are location parameters. The fitted distribution is

$$p(x) = 0.342p_{8.02}(x) + 0.330p_{7.06}(x-17) + 0.328p_{8.04}(x-32),$$

the accuracy is 0.0014. In the following figure, a "*" marks some sample probability and a "o" some fitted probabilities.

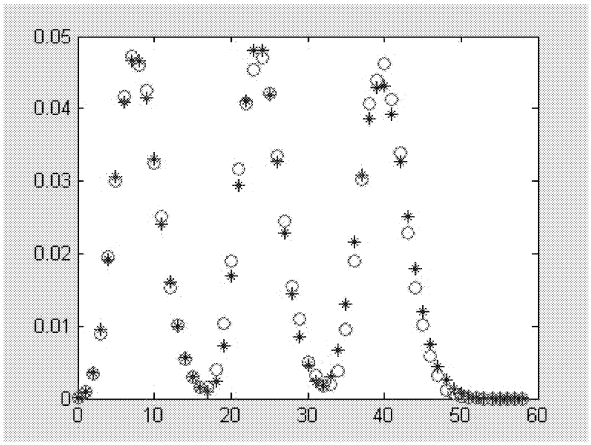


Figure 2: A Fitted Tree-Mode Poisson Distribution

Example 3, Decision between Gamma and Weibull Distribution. The sample consists in 800 realizations of a Weibull random variable with the parameter values $\alpha = 3$ and $\beta = 3$. It is tried to fit these data with a distribution function, mixed Weibull and Gamma,

$$F(x) = pF_1(x) + (1-p)F_2(x)$$

where F_1 is Weibull and F_2 is Gamma. After 400 generations, the best parameters are $p = 0.997$, $\alpha = 3.00$, $\beta = 3.02$ where the accuracy is 0.007, see also figure 3. That means, the genetic algorithm found out that the data should be fitted with the Weibull distribution, not with the Gamma distribution, and calculated the parameters.

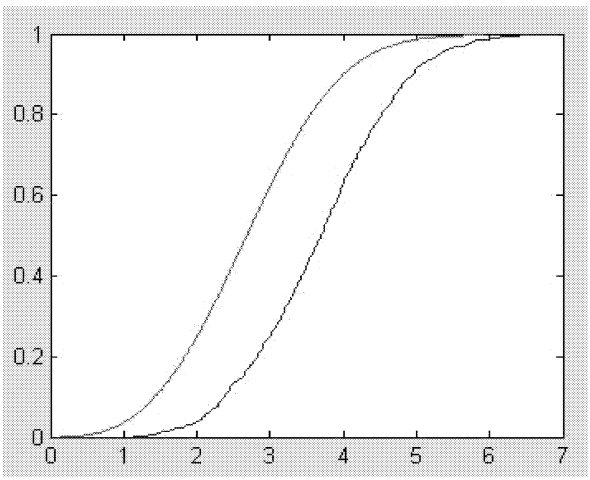


Figure 3: Decision in Favour of a Weibull Distribution

Inverse Transformation Method

The inverse transformation method is a technique for generating random variates. With a uniformly between 0 and 1 distributed ($\mathscr{U}[0,1]$-distributed) random number u, the variate x is $F^{-1}(u)$ where F^{-1} is the quasi-inverse of a CDF F,

$$F^{-1}(u) = \begin{cases} \inf\{z|F(z) \geq u\}, & u > 0, \\ \sup\{z|F(z) = 0\}, & u = 0. \end{cases}$$

This technique has some advantages. For the generation of a random variate x, right one random number u is needed, and x is monotonically increasing with u, hence positively correlated. Both features are important for some variance reduction methods.

For continuous distributions, the technique needs an inverted CDF.

For discrete distributions, the inverse transformation method is easily implemented, but not fast, in general:

Algorithm 1 Let $\{x_1,...,x_I\}$ denote the range of the distribution. The random variate is found by searching the smallest value x_j where $F(x_j) \geq u$ holds.

With linear search, this needs $O(I)$ steps, with binary search $O(\log I)$ steps.

For some discrete distributions, a fast inverse transformation method is possible. We assume a random variable X with an integer range $(i,...,j)$, probabilities $P\{X = x\} = p_x$, $x = i,...,j$, and the CDF $F(x)$ which is in fact a step function.

Theorem 1 *If there is a CDF $\tilde{F}(x)$ which has the inverse $\tilde{F}^{-1}(u)$, and for which $F(x) = \tilde{F}(x)$, $x = i,...,j$, holds, then the random variable*

$$Y = \begin{cases} \lfloor \tilde{F}^{-1}(U) \rfloor + 1, & U < 1, \\ j, & U = 1, \end{cases}$$

has the distribution $P\{Y = x\} = p_x$, $x = i,...,j$, if U is $\mathscr{U}[0,1]$-distributed.

Proof For $0 \leq U < \tilde{F}(1) = F(1) = p_1$, $Y = 1$ holds. For $\tilde{F}(x-1) < U \leq \tilde{F}(x)$, $Y = x$, $x = 2,...,j$, and $P\{\tilde{F}(x-1) < U \leq \tilde{F}(x)\} = P\{F(x-1) < U \leq F(x)\} = p_x$ hold. □

Example 4 Consider the geometric distribution with parameter p, $0 < p < 1$. The probabilities are $p_x = p(1-p)^x$, $x = 0,1,...$, and the CDF is $F(x) = 1-(1-p)^{\lfloor x \rfloor + 1}$ for $x \geq 0$. The CDF $\tilde{F}(x) = 1-(1-p)^{x+1}$ equals the CDF $F(x)$ at $x = 0,1,...$, and has the inverse $\tilde{F}^{-1}(u) = \ln(1-u)/\ln(1-p) - 1$ for $u < 1$.

Now we define a subclass of composed distributions, namely composed distributions with non-overlapping components which allow for the inverse transformation method provided that the the inverse functions of the components are available.

Definition 1 *A* composed distribution with non-overlapping components *is a composed distribution (1) where the following holds.*

1. Let $\mathscr{X}_k$ denote the the range of the random variable X_k of the component CDF $F_k(x)$, $k = 1,...,K$. For these ranges, $\{\mathscr{X}_1,...,\mathscr{X}_K\}$ is a partition of the range of the composed distribution. This partition is sorted as follows, namely for each $x \in \mathscr{X}_j$ and each $y \in \mathscr{X}_k$, $x < y \Rightarrow j < k$ holds.

2. The quasi-inverses of the components are available.

It is well known how to generate a random variate from a composed distribution (1):

1. Generate a $\mathscr{U}[0,1]$-distributed random number u_1. Apply algorithm 1 with the discrete distribution $(q_1,...,q_K)$ for generating the random variate k.

2. Generate the desired random variate x with the CDF $F_k(x)$, using one or more different, independent $\mathscr{U}[0,1]$-distributed random numbers $u_2, u_3, ...$.

For composed distributions with non-overlapping components, the inverse transformation method can be implemented as follows.

Algorithm 2

1. Generate a $\mathscr{U}[0,1]$-distributed random number u. Apply algorithm 1 with the discrete distribution $(q_1,...,q_K)$ for generating the random variate k.

2. With the $\mathscr{U}[0,1]$-distributed random number $u' = (u - (q_1 + ... + q_{k-1}))/q_k$, generate the desired random variate x from the CDF $F_k(x)$ with the inverse transformation method (we define $(q_1 + ... + q_0) = 0$).

We want to show that this algorithm is correct and is an implementation of the inverse transformation method. To this end we note that only a single random number u is used and we show that

1. u' is a realization form $\mathscr{U}[0,1]$

2. This realization is independent from the variate k

3. The generated random variates x are monotone in the random numbers u.

First, let U be a $\mathscr{U}[0,1]$-distributed random number. Define the random variables L and U' such that $q_1 + ... + q_{L-1} < U \leq q_1 + ... + q_L$, $U' = (U - (q_1 + ... + q_{L-1}))/q_L$. $0 < U' \leq 1$ follows.

Moreover, for $0 < u_0 \leq 1$

$$\begin{aligned} P\{0 < U' \leq u_0 | L = k\} &= \\ P\{0 < U' \leq u_0, L = k\}/P\{L = k\} &= \\ P\{0 < (U - (q_1 + ... + q_{k-1}))/q_k \leq u_0, L = k\}/q_k &= \\ P\{0 < (U - (q_1 + ... + q_{k-1})) \leq u_0 q_k, L = k\}/q_k &= \\ P\{q_1 + ... + q_{k-1} < U \leq q_1 + ... + q_{k-1} + u_0 q_k\}/q_k &= \\ u_0 q_k/q_k &= \\ u_0. \qquad (7) \end{aligned}$$

Hence, U' is $\mathscr{U}[0,1]$-distributed.

Secondly, from (7), $P\{U' \leq u_0, L = k\} = u_0 P\{L = k\}$ follows, hence U' and L are independent.

Thirdly, consider two random variates x and y, $x < y$, which are generated by algorithm 2 with the $\mathscr{U}[0,1]$-distributed random numbers u and v. Then $u < v$ holds which can be seen as follows. Either, x and y are elements of the same subrange $\mathscr{X}_k$. Then $u < v$ holds because x and y are generated with the inverse transformation method using the CDF $F_k(x)$. Or they are from different subranges, $x \in \mathscr{X}_k$ and $y \in \mathscr{X}_l$. Then, $k < l$ holds according to the definition of composed distribution with non-overlapping components, and $q_1 + ... + q_{k-1} < u \leq q_1 + ... + q_k$ and $q_1 + ... + q_{k-1} + q_k + ... + q_{l-1} < v \leq q_1 + ... + q_l$. Hence, $u < v$ holds.

Thus we have

Theorem 2 *Algorithm 2 implements the inverse transformation method.*

Before closing this subsection about the inverse transformation method, we consider two standard distributions which are suitable as components of composed distributions with or without non-overlapping components, the single point distribution and the discrete trapezoid distribution which are not very common in simulation.

Definition 2 *A random variable X is* single point distributed *if the probability mass is concentrated at a single value* $x^{(single)}$, $P\{X = x^{(single)}\} = 1$.

Definition 3 *A random variable X has a* discrete trapezoid distribution *if the probabilities are*

$$P\{X = x\} = p_0 + \Delta(x - x_0),\ x = x_0, ..., x_0 + l - 1. \qquad (8)$$

As parameters, we use x_0, $l \geq 1$, integer, and a, $0 \leq a \leq 1$, real. For $l > 1$, we set $p_0 = 2a/l$ and $\Delta = 2(1-2a)/l/(l-1)$. For $l = 1$, a must be $1/2$, and we set $\Delta = 0$. Then all probabilities sum to 1 and are non-negative.

Remark The probabilities are linear in x, the parameter a determines the slope which is positive for $a < 1/2$ and negative for $a > 1/2$. For $a = 1/2$, this is the discrete uniform distribution.

The discrete trapezoid distribution is invertible, see theorem 1. For $\Delta > 0$,

$$F^{-1}(u) = x_0 + \begin{cases} \left\lfloor -\frac{p_0}{\Delta} + \frac{1}{2} + \text{sign}(\Delta)\sqrt{(\frac{p_0}{\Delta} - \frac{1}{2})^2 + \frac{2u}{\Delta}} \right\rfloor, & u < 1, \\ l - 1, & u = 1, \end{cases}$$

and for $\Delta = 0$,

$$F^{-1}(u) = x_0 + \begin{cases} \lfloor lu - 1 \rfloor + 1, & u < 1, \\ l - 1, & u = 1. \end{cases}$$

4 EXAMPLE: MODELLING A MEASURED DISCRETE SAMPLE

In this section, we consider a sample of measured packet sizes in an IP-network, see Klemm, Lindemann, and Lohmann (2002). We model the empirical data with a composed distribution, since the distribution has a quite irregular shape. The range is 32,...,1500 with 52 holes, some values occur much more often than the values surrounding them.

For a first analysis, we plotted the empirical probabilities in different scales. Thus we found the values which are not existent or which are prevalent. We model the prevalent value probabilities with 26 single point distributions. The following figure shows the empirical probabilities with logarithmic scale (but "0" means value zero). The circled points are the identified prevalent values.

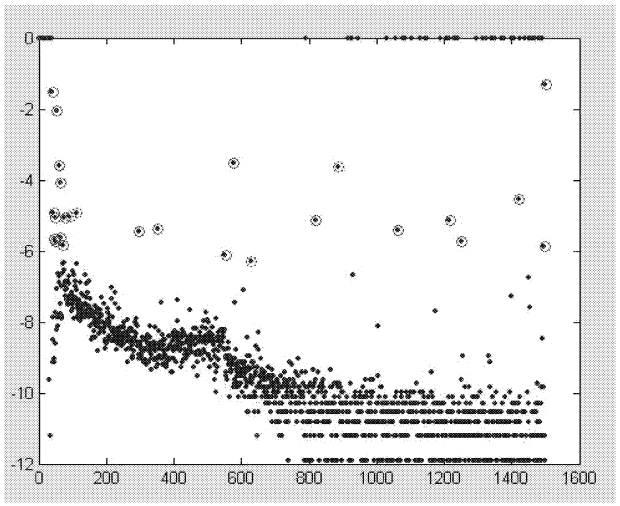


Figure 4: Empirical Probabilities in Logarithmic Scale

For all other range values, we provided six discrete trapezoid distributions, over six subranges which constitute a partition of the range. Thus we get a composed distribution with 32 components. Since the 26 prevalent values are elements of the subranges, the composed distribution is not with non-overlapping components.

Prevalent probabilities can be identified looking into the plots. Actually, we selected them algorithmically. Broadly speaking, we compared all probabilities with a moving average, and if they are considerably bigger (twofold or more), we marked them as prevalent, if they are also bigger than 0.0013.

For the discrete trapezoid distributions, we looked into the plots and defined points where we had the impression that the slope should change: They became the boundaries of the

subranges with different trapezoid distributions: 32, 75, 200, 350, 550, 800, 1500.

The fitted composed distribution, based on 40000 sample points, has a very good accuracy of 0.0002. The following figures also exhibit the accuracy.

The next figure presents the empirical CDF and the fitted CDF which is augmented by 0.1 in order to separate the graphs.

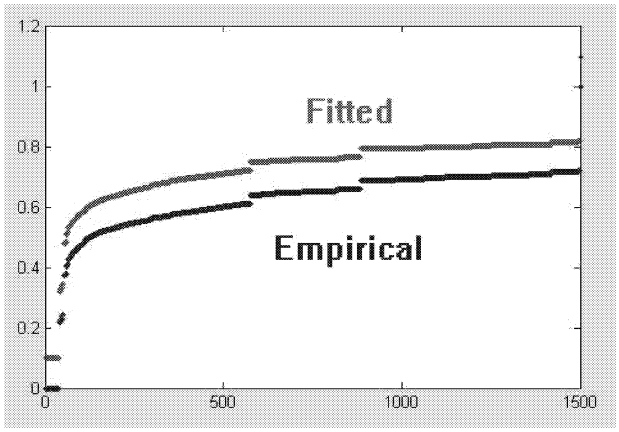


Figure 5: Empirical CDF and Fitted CDF plus 0.1

Figure 6 is a P-P plot with the fitted CDF values versus the empirical CDF values. For perfectly equal values, all points would lie on a straight line with an intercept of 0 and a slope of 1.

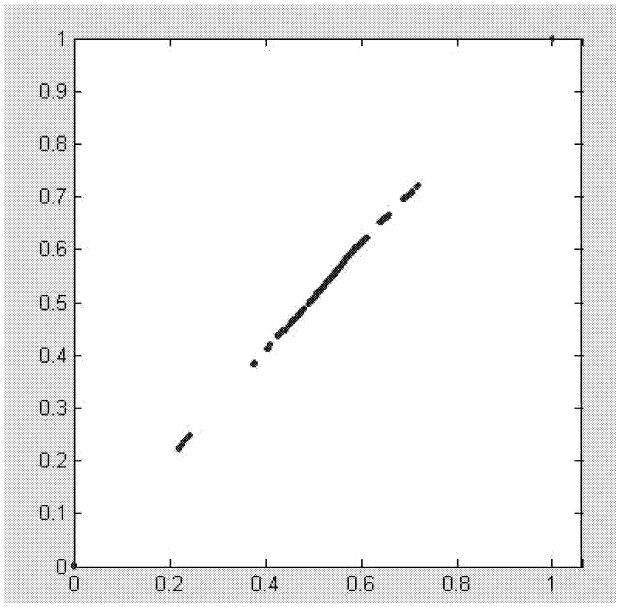


Figure 6: Probability-Probability Plot

Now we present two figures with differences, first beween the fitted CDF values and the empirical CDF values, and then beween the fitted probabilities and the empirical probabilities.

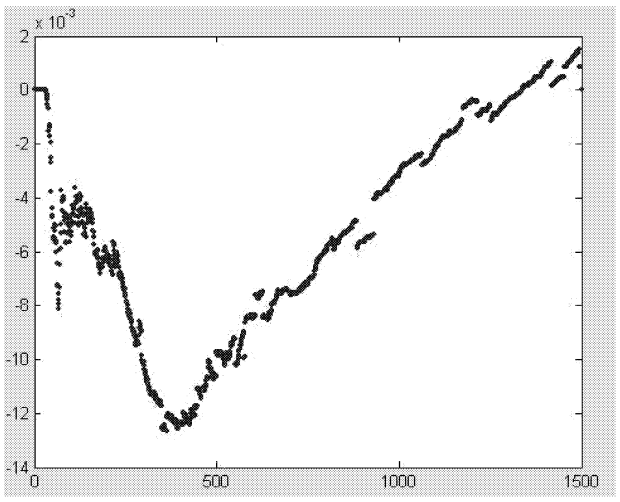


Figure 7: Differences between Fitted and Empirical CDF Values

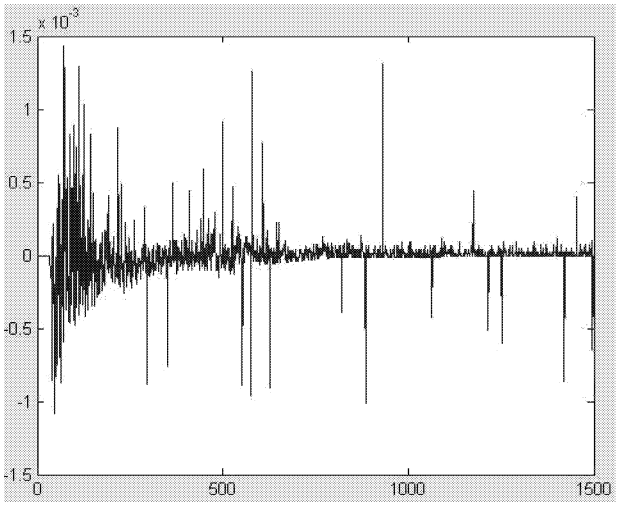


Figure 8: Differences between Fitted and Empirical Probabilities

CONCLUSION

We presented a new technique for modelling and fitting complex input distributions for stochastic models which has some advantages compared to common techniques. These ideas are intended to be brought in a fitting tool.

REFERENCES

Bäck, T., D. Fogel, and Z. Michalewicz. (Eds.) 1997. *Handbook of evolutionary computation*. Bristol, New York: Oxford University Press.

Chipperfield, A., P. Fleming, H. Pohlheim, and C. Fonseca. *Genetic algorithm toolbox for use with MATLAB*. http://www.shef.ac.uk/~gaipp/ga-toolbox/.

Davis, L. (Ed.) 1991. *Handbook of genetic algorithms*. New York: Van Norstrand Reinhold.

Goldberg, D. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading,MA: Addison-Wesley.

Klemm, A., C. Lindemann, and M. Lohmann. 2002. Traffic modeling of IP networks using the batch Markovian arrival process. In *Proceedings of the 12th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation*, 92–110. London: Springer. LNCS 2324.

Law, A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*. third ed. New York: McGraw-Hill.

Matlab. <www.mathworks.com>.

Strelen, J. C. 2003. The genetic algorithm is useful to fitting input probability distributions for simulation models. In *Proceedings of the Business and Industry Symposium - ASTC 2003*, ed. M. Ades and L. Deschaine, 8–13. San Diego: Society for Computer Simulation.

Strelen, J. C., and F.Nassaj. 2007. Analysis and generation of random vectors with copulas. In *To appear in the Proceedings of the 2007 Winter Simulation Conference, Washington*, ed. S.G.Henderson, B.Biller, M.-H.Hsieh, J.Shortle, J.D.Tew, and R.R.Barton.

AUTHOR BIOGRAPHY

JOHANN CHRISTOPH STRELEN received the Dipl.-Math. and Dr. rer. nat. degrees in mathematics and the Habilitation degree in Computer Science from the Technische Hochschule Darmstadt, Germany. He is Professor of Computer Science at the university of Bonn, Germany. His research interests include performance evaluation, distributed systems, and simulation. His e-mail address is <strelen@cs.uni-bonn.de> and his web address is <web.cs.uni-bonn.de/IV/strelen/strelen_en.html>.

EFFECTIVE APPROXIMATION OF ZERO-VARIANCE SIMULATION IN A RELIABILITY SETTING

Pierre L'Ecuyer

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal (Québec), H3C 3J7, CANADA

Bruno Tuffin

IRISA-INRIA, Campus de Beaulieu
35042 RENNES Cedex, FRANCE

ABSTRACT

For every stochastic simulation model, there is in theory a way of changing the probability laws that drive the system so that the resulting IS estimator has zero variance. This optimal estimation scheme is generally impractical to implement, but it can be possible to approximate it in an effective way. When the model is described by a discrete-time Markov chain that evolves up to some random stopping time, the zero-variance change of measure can be written exactly in terms of a value function that gives the expected cost-to-go from any state of the chain, so it can be approximated by approximating this value function. We detail this approach and show how it can be effectively used to estimate the reliability of a highly-reliable multicomponent system with Markovian behavior. In our implementation, we start with a very simple crude approximation, use it in a first-order IS scheme to obtain a better approximation at a few selected states, interpolate in between, and use this interpolation in our final (second-order) IS scheme. In numerical illustrations, our approach outperforms the popular IS heuristics previously proposed for this class of problems.

INTRODUCTION

Any discrete-event simulation model can be represented as a discrete-time Markov chain (DTMC), say $\{Y_j,\, j \geq 0\}$, with very large (usually infinite and high-dimensional) state space $\mathscr{Y}$. By putting enough information in the state, this chain can also be defined so that the simulation terminates when (and only when) the chain reaches a given (fixed) subset of the state space, say $\Delta \subset \mathscr{Y}$. Suppose there is a cost $c(y,y')$ whenever the chain goes from state y to state y' at any given step, and we want to estimate $\mu(y_0) = \mathbb{E}[X \mid Y_0 = y_0]$, where X is the total cost until termination, and the conditioning means that the chain starts from a given initial state $Y_0 = y_0$.

The standard Monte Carlo method estimates $\mu(y_0)$ by simulating n independent copies of the DTMC under its original probability law, and averages the n copies of X (Fishman 1996, Law 2007). A confidence interval on $\mu(y_0)$ is usually computed by assuming that this average is approximately normally distributed (relying on the central-limit theorem) and estimating its variance by the sample variance of the n copies of X, divided by n. There are many situations, however, where the required accuracy calls for a very large value of n, sometimes so large that the simulation time becomes excessive. This is particularly true in the context of rare-event simulation, where certain events having a large influence on the expected cost occur very rarely (Heidelberger 1995, Juneja and Shahabuddin 2006). If an important rare event happens only once every billion simulation on average, for example, then n must be at least several billion before we start to obtain an estimator with some meaning.

Variance reduction methods have been designed to solve or alleviate this type of problem (Fishman 1996). In this paper, we focus on one of these methods, called *importance sampling* (IS), which consists in changing the probability laws that drive the system, and multiplying the final estimator by an appropriate correction factor to recover an unbiased estimator of the original expectation (Glynn and Iglehart 1989, Juneja and Shahabuddin 2006). This correction factor turns out to be the likelihood ratio (or Radom-Nikodym derivative) between the old and new probabilities. A major difficulty in applying this method is to figure out how to change the probabilities so that the IS estimator has much smaller variance than the original one.

In theory, there is actually a way of changing the transition probabilities of the chain so that the new estimator has zero variance! In a nutshell, it suffices to multiply the transition probability (or density) from the current state y to any other state y' by the ratio of the expected total cost from now on if we go next to state y', divided by the (unconditional) expected total cost from now on. Details will be given in the next section. Special cases are examined in Booth (1987), Kollman et al. (1999), Juneja and Shahabuddin (2006), for example. So if we would know the expected total cost until absorption as a function of the initial state y, $\mu(y)$, we would be able to define a zero-variance sampling scheme. This

function $\mu(y)$ is called the *cost-to-go* function, or Bellman function, in dynamic programming. The catch is that in practice, we do not know this function. If we knew it, there would be no need for simulation!

But not all hope is lost. By obtaining a rough-cut approximation of the cost-to-go function μ, and using it in place of μ to approximate the zero-variance change of measure, we may get an IS scheme that achieves a significant variance reduction compared with standard Monte Carlo.

In this paper, we first explain in detail (in the next section) how to define the zero-variance sampling scheme for a DTMC. Then, in the following section, we select a specific class of problems in which the goal is to estimate the reliability of a highly-reliable multicomponent system with Markovian behavior, and show how to approximate the zero-variance change of measure for that class of problems. The models we consider are known in the literature under the name of highly-reliable Markovian systems (HRMS) (Shahabuddin 1994b, Cancela et al. 2002, Nakayama and Shahabuddin 2004). According to our numerical examples in the last section, the new method is much more effective than the IS heuristics previously proposed for this class of problems and are applicable to large problem instances.

MARKOV CHAIN MODEL AND ZERO-VARIANCE SAMPLING

We consider a discrete-time Markov chain (DTMC) $\{Y_j,\, j \geq 0\}$ with general state space $\mathscr{Y}$, transition kernel P, and nonnegative one-step cost function $c : \mathscr{Y}^2 \to [0,\infty)$. When the chain is in state $Y_{j-1} = y \in \mathscr{Y}$, the next state Y_j obeys a probability law defined by $\mathbb{P}[Y_j \in B \mid Y_{j-1} = y) = P(B \mid y)$ for all (measurable) $B \subseteq \mathscr{Y}$, and a transition cost $c(y,Y_j)$ is incurred. The state space contains a set of absorbing states $\Delta \subset \mathscr{Y}$ from which the transition cost is zero: $P(\{y\} \mid y) = 1$ and $c(y,y) = 0$ for all $y \in \Delta$. Let $\tau = \inf\{j : Y_j \in \Delta\}$, the number of steps until absorption,

$$X = \sum_{j=1}^{\tau} c(Y_{j-1}, Y_j),$$

the total cost until absorption, and

$$\mu(y) = \mathbb{E}[X \mid Y_0 = y],$$

the expected total cost when starting in state y. We assume that $\mathbb{E}[\tau \mid Y_0 = y] < \infty$ and $\mu(y) < \infty$ for all $y \in \mathscr{Y}$. We also assume implicitly that the required measurability conditions hold so that all the expressions introduced here and later are well defined.

The function $\mu : \mathscr{Y} \to [0,\infty)$ satisfies the recurrence

$$\begin{aligned} \mu(y) &= \mathbb{E}[c(y,Y_1) + \mu(Y_1) \mid Y_0 = y] && (1) \\ &= \int_{\mathscr{Y}} [c(y,y_1) + \mu(y_1)]\, dP(y_1 \mid y) && (2) \end{aligned}$$

for all $y \in \mathscr{Y}$.

We consider changing the transition kernel P for another kernel Q such that $Q(B \mid y) > 0$ whenever $\int_B [c(y,y_1) + \mu(y_1)]\, dP(y_1 \mid y) > 0$. The estimator X is then replaced by

$$X_{\text{is}} = \sum_{j=1}^{\tau} c(Y_{j-1}, Y_j) \prod_{i=1}^{j} L(Y_{i-1}, Y_i), \tag{3}$$

where $L(Y_{i-1}, Y_i) = (dP/dQ)(Y_i \mid Y_{i-1})$ and dP/dQ is the Radon-Nikodym derivative of P with respect to Q (Glynn and Whitt 1989, Heidelberger 1995, Bucklew 2004, Juneja and Shahabuddin 2006). Let $\mathbb{E}_{Q,y}$ and $\text{Var}_{Q,y}$ denote the expectation and variance operators under the measure (transition kernel) Q, from initial state $Y_0 = y$. We have

$$\mathbb{E}_{Q,y}[X_{\text{is}}] = \mu(y),$$

i.e., X_{is} is an unbiased estimator of $\mu(y)$ under the new kernel Q. The variance of X_{is} under Q, for $Y_0 = y$, is

$$\begin{aligned} V(y) &\stackrel{\text{def}}{=} \text{Var}_{Q,y}[X_{\text{is}}] \\ &= \text{Var}_{Q,y}[\mathbb{E}_{Q,y}[X_{\text{is}} \mid Y_1]] + \mathbb{E}_{Q,y}[\text{Var}_{Q,y}[X_{\text{is}} \mid Y_1]] \\ &= \text{Var}_{Q,y}[(c(y,Y_1) + \mu(Y_1))L(y,Y_1)] \\ &\quad + \mathbb{E}_{Q,y}[L^2(y,Y_1)V(Y_1)] \\ &= \mathbb{E}_{Q,y}[(c(y,Y_1) + \mu(Y_1))^2 L^2(y,Y_1)] - \mu^2(y) \\ &\quad + \mathbb{E}_{Q,y}[L^2(y,Y_1)V(Y_1)] \\ &= \mathbb{E}_{Q,y}[((c(y,Y_1) + \mu(Y_1))^2 + V(Y_1))L^2(y,Y_1)] \\ &\quad - \mu^2(y). \end{aligned}$$

Suppose now that we choose the measure Q in a way that for each y such that $\mu(y) > 0$,

$$dQ(y_1 \mid y) = dP(y_1 \mid y)(c(y,y_1) + \mu(y_1))/\mu(y). \tag{4}$$

By integrating with respect to y_1, we easily see that $Q(\cdot \mid y)$ thus defined is a probability measure (it integrates to 1). If $\mu(y) = 0$, we simply take $Q(\cdot \mid y) = P(\cdot \mid y)$. With this choice of Q, whenever $dQ(y_1 \mid y) > 0$, we have

$$L(y,y_1) = \begin{cases} \mu(y)/(c(y,y_1) + \mu(y_1)) & \text{if } \mu(y) > 0, \\ 1 & \text{if } \mu(y) = 0. \end{cases} \tag{5}$$

For state pairs (y, y_1) for which $dQ(y_1 \mid y) = 0$, $L(y, y_1)$ may be undefined, but this does not matter because $L(y, y_1)$ will then never occur in the estimator. If (5) holds, then

$$\mathbb{E}_{Q,y}[(c(y,Y_1)+\mu(Y_1))^2 L^2(y,Y_1)] = \mu^2(y)$$

and we have the simplification:

$$\begin{aligned} & V(y) \\ = \;& \mathbb{E}_{Q,y}[((c(y,Y_1)+\mu(Y_1))^2 + V(Y_1))L^2(y,Y_1)] - \mu^2(y) \\ = \;& \mathbb{E}_{Q,y}[V(Y_1)L^2(y,Y_1)]. \end{aligned}$$

Applying induction, we obtain

$$V(y) = \mathbb{E}_{Q,y}\left[V(Y_\tau)\prod_{j=1}^{\tau} L(Y_{i-1}, Y_i)\right] = 0$$

because $V(Y_\tau) = 0$. We conclude that a change of measure that satisfies (5) gives a zero-variance estimator of $\mu(y)$ for any $y \in \mathscr{Y}$.

As a special case, suppose that the state space $\mathscr{Y}$ is finite, so that the DTMC has discrete transition probabilities $p(y_1 \mid y) = \mathbb{P}[Y_1 = y_1 \mid Y_0 = y]$, which we replace by new probabilities $q(y_1 \mid y) = \mathbb{Q}[Y_1 = y_1 \mid Y_0 = y]$, such that $q(y_1 \mid y) > 0$ whenever $[c(y,y_1) + \mu(y_1)]p(y_1 \mid y) > 0$. In this case, we have $L(y,y_1) = p(y_1 \mid y)/q(y_1 \mid y)$. To satisfy (5), it suffices to choose the probabilities

$$q(y_1 \mid y) = p(y_1 \mid y)(c(y,y_1) + \mu(y_1))/\mu(y) \qquad (6)$$

if $\mu(y) > 0$ and $q(y_1 \mid y) = p(y_1 \mid y)$ otherwise.

Since μ is generally unknown, the idea is to approximate it by some function v that can easily be computed during the simulation. This function can also be modified or "learned" along the way, from the results of the simulation runs. For each run, we use the function v that we have so far (which is our best estimate of μ), and plug it in (6) or (4) in place of μ to define the change of probabilities for the IS estimator. That is, in the case of a finite-state Markov chain, we use

$$q(y_1 \mid y) = p(y_1 \mid y)(c(y,y_1) + v(y_1))/v(y) \qquad (7)$$

if $v(y) > 0$ and $q(y_1 \mid y) = p(y_1 \mid y)$ otherwise.

One class of methods to approximate μ collect information from the simulation to "learn" about $\mu(y)$ for each state $y \in \mathscr{Y}$. Ahamed et al. (2006) do this using a stochastic approximation approach to update the individual entries $\mu(x)$. Booth (1982; 1987) has also been using this type of approach to learn μ for both splitting and IS; this is implemented in the Los Alamos simulation software. However, this type of approach has limitations: for real-life problems for which Monte Carlo methods are used, the state space is usually much too large to think of storing an approximation of $\mu(y)$ for each state y. For this reason, we will not consider it any further in this paper. What we propose instead is to construct a leaner approximation of μ, that can be computed quickly at any state y, and which requires much less storage than directly estimating $\mu(y)$ for each state y.

A MODEL OF HIGHLY RELIABLE MARKOVIAN SYSTEM

We consider an HRMS model as in Shahabuddin (1994b;a), Nakayama and Shahabuddin (2004), among others. The system has c types of components, with n_i identical components of type i, for $i = 1, \ldots, c$. Each component is either in a failed state or an operational state. The *state of the system* is represented by a vector $y = (y^{(1)}, \ldots, y^{(c)})$, where $y^{(i)}$ is the number of *failed* components of type i. Thus, we have a finite state space $\mathscr{Y}$ of cardinality $(n_1+1)\cdots(n_c+1)$. This system is assumed to evolve as a continuous-time Markov chain (CTMC), with jump rate $\lambda(y,y')$ from state y to state y'. We suppose that $\mathscr{Y}$ is partitioned in two subsets $\mathscr{U}$ and $\mathscr{F}$, where $\mathscr{U}$ is a decreasing set (i.e., if $y \in \mathscr{U}$ and $y \geq y' \in \mathscr{S}$, then $y' \in \mathscr{U}$) that contains the state $\mathbf{0} = (0, \ldots, 0)$ in which all the components are operational.

The CTMC has an embedded DTMC with transition probabilities $p(y,y')$. We assume that state $\mathbf{0}$ can be reached from any state $y \in \mathscr{U}$. Define $\mu(y) = \mathbb{P}[\tau_{\mathscr{F}} < \tau_{\mathbf{0}} \mid Y_0 = y]$, where $\tau_{\mathscr{F}} = \inf\{j \geq 0 : Y_j \in \mathscr{F}\}$ and $\tau_{\mathbf{0}} = \inf\{j > 0 : Y_j = \mathbf{0}\}$. Our goal is to estimate $\mu(\mathbf{0})$. Note that $\mu(y) = 1$ when $y \in \mathscr{F}$. Estimating $\mu(\mathbf{0})$ is required in the estimation of several performance measures, such as the mean time to failure, via a regenerative approach (Shahabuddin 1994a).

This estimation can be especially difficult when $\mu(\mathbf{0})$ is extremely small, which occurs frequently in the case of highly reliable systems. Then, the transitions that correspond to component failures have very small probabilities, while those that correspond to repairs are not so small. In theoretical analysis of such systems, the failure probabilities are often parameterized by some parameter ε so that these probabilities converge to 0 when $\varepsilon \to 0$ while the repair probabilities remain bounded, and one is interested in the asymptotic behavior of the estimator when $\varepsilon \to 0$. One usually seeks estimators with the highly desirable property of *bounded relative error* (BRE) as a function of ε, which means that the variance divided by the square of $\mu(\mathbf{0})$ remains bounded when $\varepsilon \to 0$.

As a rough-cut approximation of the function μ in this model, we start with the following. For any state $y \in \mathscr{U}$, let $\Gamma(y)$ be the set of all paths $\pi = (y = y_0 \to y_1 \to \cdots \to y_k)$ going from state y to the set $\mathscr{F}$, where $y_j \in \mathscr{U} \setminus \{\mathbf{0}\}$ for $j = 1, \ldots, k-1$, $p(y_{j-1}, y_j) > 0$ for $j = 1, \ldots, k$, and $y_k \in \mathscr{F}$. Each path $\pi \in \Gamma(y)$ has original probability

$$p(\pi) = \prod_{j=1}^{k} p(y_{j-1}, y_j)$$

and we have that $\mu(y) = \sum_{\pi \in \Gamma(y)} p(\pi)$. However, the latter sum is usually too complicated to compute in practice, because it involves too many paths (there is often an infinite number of paths, because the paths may contain an unlimited number of cycles within $\mathscr{U}$). A very crude estimate is to replace the sum by the maximum; i.e., approximate $\mu(y)$ by its lower bound

$$v_0(y) = \max_{\pi \in \Gamma(y)} p(\pi).$$

Computing this $v_0(y)$ amounts to computing the shortest path from y to $\mathscr{F}$, where the length of the directed link from y' to y'' is $-\log p(y', y'')$ for any pair of states (y', y''), or equivalently the longest path where the length of a path is its probability (the product of probabilities of its one-step transitions). It can be computed cheaply even for complicated structures. When such a shortest path from some state y is computed, one can store it in a hash table together with all other shortest paths that are computed simultaneously or have been computed previously. Whenever we need $v_0(y)$ for some state y, we first check the hash table to see if it has already been computed.

This v_0 would do well in the cases where a single path dominates the sum, which sometimes occurs when all transitions toward $\mathscr{F}$ have a very small probability. But this lower bound on $\mu(y)$ could still underestimate the true value by a significant factor. An easy improvement is to take the sum over the most dominant paths (only a few of them) instead of just considering the single most dominant one. These dominant paths can often be selected by exploiting our knowledge of the structure of the system.

In L'Ecuyer and Tuffin (2006), we had good luck with the following simple type of correction (this was used there in a splitting algorithm): estimate $\mu(\mathbf{0})$ in preliminary runs with some initial IS strategy, and compute the exponent α such that $(v_0(\mathbf{0}))^\alpha$ equals this estimate. Then, replace the estimate $v_0(y)$ by $v_1(y) = (v_0(y))^\alpha$ for all $y \in \mathscr{U}$. This function v_1 matches μ for $y \in \mathscr{F}$ and matches its estimate at $y = \mathbf{0}$. In between, it uses an exponential interpolation, motivated by the idea that $\mu(y)$ typically decreases exponentially (roughly) when y gets away from $\mathscr{F}$.

This idea could be refined. One possibility is to replace α by a state-dependent correction exponent $\alpha(y)$. Here we shall consider the following form for $\alpha(y)$:

$$\alpha(y) = 1 + [\alpha(\mathbf{0}) - 1] \frac{\log v_0(y)}{\log v_0(\mathbf{0})},$$

where $\alpha(\mathbf{0})$ is the value of α as in the previous paragraph. The rationale for this form is that the correction exponent is needed usually because $v_0(y)$ accounts for only a few paths and disregards many other ways of reaching $\mathscr{F}$ from y. Since the set of paths leading to $\mathscr{F}$ is generally richer when we are farther from $\mathscr{F}$, it appears sensible to have a correction exponent that changes progressively from 1 when we are very close to $\mathscr{F}$, to $\alpha(\mathbf{0})$ when we are in state $\mathbf{0}$, and reflects the "distance" to $\mathscr{F}$ for the states in between. We denote the resulting approximation by $v_2(y) = (v_0(y))^{\alpha(y)}$.

Among other possibilities, instead of estimating $\mu(y)$ only at $y = \mathbf{0}$, we can estimate it directly, via IS, over a finite subset of states $\mathscr{E} \subset \mathscr{U}$, in preliminary runs. For example, in an HRMS model, $\mathscr{E}$ could be the set of states where no more than one component is failed, or the set of states where no more than two components are failed (depending on the structure and size of the model). For each state $y \in \mathscr{E}$, we define $v_3(y)$ as the direct estimate of $\mu(y)$ and we compute $\alpha(y)$ such that $(v_0(y))^{\alpha(y)} = v_3(y)$. For each state $y \in \mathscr{U} \setminus \mathscr{E}$, we may interpolate exponentially as follows: We can select a state $y' \in \mathscr{E}$ such that $y' \le y$, and define $v_3(y) = (v_0(y))^{\alpha(y')}$ (the selection is arbitrary and could be problem-dependent). Or we may select *all* states $y' \in \mathscr{E}$ such that $y \ne y' \le y$, and define $v_4(y)$ as the average of the corresponding values of $(v_0(y))^{\alpha(y')}$. Yet another possibility is to estimate a *single* exponent α for all states y, based on all available information, e.g., by least-squares regression. As an extreme case, taking $\mathscr{E} = \mathscr{U}$ means that we have a direct estimate of $\mu(y)$ for all states $y \in \mathscr{U}$, and no interpolation is needed. Then, we are back to an estimator similar to that of Ahamed et al. (2006), depending on how the estimation is done (these authors change the measure dynamically at each step, at the same time as they update their estimates of $\mu(y)$). Our proposal is a matter of compromise between this extreme case and just taking v_0 (the other extreme).

All these possibilities would deserve further analysis and empirical comparison in realistic examples. In the next section, we compare some of them with the best known IS heuristics, on a parameterized example.

Various heuristics have actually been proposed to select the IS change of measure for the type of HRMS setting considered here (Shahabuddin 1994b, Nakayama 1996, Cancela et al. 2002, Tuffin 2004). These methods do not directly attempt to approximate the zero-variance IS. Their aim is

to increase the probabilities of the transition failures, so that we hit $\mathscr{F}$ before $\mathbf{0}$ more frequently. The best known are probably *simple failure biasing* (SFB) and *balanced failure biasing* (BFB), developed by Shahabuddin (1994b). For each state $y \in \mathscr{U} \setminus \{\mathbf{0}\}$, SFB and BFB increase the probability that the next transition is a failure to a fixed constant $0 < \rho < 1$. Within the subset of failure transitions, the individual probabilities are taken proportional to the original ones for SFB and uniform over all transitions for BFB. Within the set of repair transitions, they are taken proportional to the original ones for both schemes. Under certain conditions on the system, BFB has been proved to yield an IS estimator with the BRE property. The standard MC estimator does not enjoy this property.

Alexopoulos and Shultes (2001) have proposed a variation called SBLR, in which the change of measure is made in a way that over any cycle in the sequence of states visited during the simulation, the cumulated likelihood ratio remains bounded when the failure probabilities converge to zero. Some improvements are also proposed that use structural information by identifying events on shortest paths to failure, and pushing more toward those events. We do not consider these versions here.

NUMERICAL EXAMPLE

Consider a system with $c = 3$ component types, with $n_1 = n_2 = n_3$. Any failed component has an exponentially distributed repair time with rate 1, and an exponentially distributed time to failure with rate λ_i for components of type i, where $\lambda_1 = \varepsilon$, $\lambda_2 = 1.5\varepsilon$, and $\lambda_3 = 2\varepsilon^2$, for some parameter ε. The system is down whenever fewer than two components of anyone type are operational. The system starts in state $\mathbf{0}$ in which all components are operational, and we want to estimate the probability $\mu(\mathbf{0})$ that it reaches the set $\mathscr{F}$ of failure states before returning to state $\mathbf{0}$. We will experiment with different values of the two parameters (n_i, ε), given in the first two columns of Table 1.

For this example, we define $v_0(y)$ as follows. For each component type i, we consider the path that goes from y to $\mathscr{F}$ whose only transitions are failures of components of type i. There are three such paths for each state $y \in \mathscr{U}$, and their probabilities $p(\pi)$ are very easy to compute. We simply define $v_0(y)$ as the sum of their probabilities.

The methods we try and compare are BFB, SBLR, and our proposed IS method based on zero-variance approximation, using the function v_0 just described, and its modifications v_1 and v_2 defined earlier. We will denote these methods by IS(v_0), IS(v_1), IS(v_2), respectively. For each parameter set, Table 1 gives our best estimate of $\mu(\mathbf{0})$, obtained from a very large number of simulation runs with our IS strategies (these numbers are accurate at least for the digits given in the table), the rough-cut approximation $v_0(\mathbf{0})$ of $\mu(\mathbf{0})$, and the estimate obtained from $n = 2^{20}$ (approximately one million) independent simulation runs, by each of the five methods. Table 2 shows the empirical variances computed from those n runs, for each method. The table also contains the squared mean, so that one can quickly get an idea of the relative variance (the squared relative error, i.e., the variance divided by the square mean).

In those tables, the entries in parentheses (and in red) are empirical means and variances that clearly underestimate their exact counterparts by a large factor. The red entries for the empirical means are actually even lower (by a large factor) than the crude lower bound $v_0(\mathbf{0})$. When we have serious underestimation for the mean, then we have it for the variance as well. The usual explanation is that certain types of paths that have an important relative contribution to the mean are given a too small probability by the IS heuristic, and never occur in the sample, so their contribution is totally missed by the estimator. This reduces both the empirical mean and empirical variance (but not the true variance). If one of these paths would occur, it would potentially have a huge contribution, due to a large likelihood ratio. Interestingly, the entry in brackets (in green) underestimates the exact mean by a significant factor. All these problematic entries are for the BFB and SBLR heuristics, and things generally worsen when $\mu(\mathbf{0})$ gets smaller and n_i gets larger. These heuristics were designed to cope with with very low failure rates for the components (very small ε), but are not doing very when the paths to $\mathscr{F}$ have a large number of transitions (i.e., when n_i is large, in our example).

With our proposed approach, using v_0 as an approximation already gives much better results than BFB and SBLR, in the sense that we at least get the right order of magnitude for all parameter values. The adjustment v_1 does not provide much improvement over v_0 in this example, whereas v_2 does provide a significant improvement. This seems to confirm the idea that the exponential correction should take into account the distance to failure. This appears to be especially true when n_i is small, in which case we need much less corrections for the states that are very close to failure than for the initial state (compare the results of v_2 with the other ones when $n_i = 3$).

With any of the three variants (v_0, v_1, and v_2), we are able to estimate very small probabilities (smaller than 10^{-55} in our example) quite accurately with a reasonably small number of simulation runs. For example, for $n_i = 12$ and $\varepsilon = 0.001$, if we take the average over n simulation runs with v_2, the relative error is $\sqrt{9.4 \times 10^{-56}}/(3.9 \times 10^{-28}\sqrt{n}) \approx 0.786/\sqrt{n}$, so we only need $n = (78.6/x)^2$ to get $x\%$ relative

Table 1: Parameter sets and estimates of $\mu(\mathbf{0})$ with each method.

n_i	ε	$\mu(\mathbf{0})$	$v_0(\mathbf{0})$	BFB	SBLR	IS(v_0)	IS(v_1)	IS(v_2)
3	0.001	2.6×10^{-3}	1.3×10^{-3}	2.7×10^{-3}	2.6×10^{-3}	2.6×10^{-3}	2.6×10^{-3}	2.6×10^{-3}
6	0.01	1.8×10^{-7}	3.4×10^{-8}	1.9×10^{-7}	$[9.9\times10^{-7}]$	1.8×10^{-7}	1.8×10^{-7}	1.8×10^{-7}
6	0.001	1.7×10^{-11}	3.4×10^{-12}	1.8×10^{-11}	(1.8×10^{-16})	1.7×10^{-11}	1.7×10^{-11}	1.7×10^{-11}
12	0.1	6.0×10^{-8}	3.2×10^{-9}	4.8×10^{-8}	1.3×10^{-8}	6.0×10^{-8}	6.2×10^{-8}	6.7×10^{-8}
12	0.001	3.9×10^{-28}	3.5×10^{-29}	(1.8×10^{-40})	(2.9×10^{-45})	3.9×10^{-28}	3.9×10^{-28}	3.9×10^{-28}

Table 2: Empirical variances based on $n = 2^{20}$ independent runs.

n_i	ε	$(\mu(\mathbf{0}))^2$	α	BFB	SBLR	IS(v_0)	IS(v_1)	IS(v_2)
3	0.001	6.8×10^{-6}	0.906	1.8×10^{-2}	8.0×10^{-3}	6.5×10^{-4}	2.7×10^{-3}	9.3×10^{-9}
6	0.01		0.903	6.3×10^{-11}	(4.5×10^{-16})	2.0×10^{-14}	1.2×10^{-14}	7.7×10^{-15}
6	0.001		0.939	8.8×10^{-19}	(2.0×10^{-26})	1.2×10^{-23}	1.1×10^{-23}	7.6×10^{-24}
12	0.1		0.851	8.1×10^{-10}	1.7×10^{-10}	1.6×10^{-10}	2.9×10^{-10}	1.5×10^{-11}
12	0.001		0.963	(3.2×10^{-74})	(3.5×10^{-84})	1.4×10^{-55}	9.3×10^{-56}	9.4×10^{-56}

error. For example, $n = 61$ suffice to obtain 10% relative error. For $n_i = 12$ and $\varepsilon = 0.1$, on the other hand, the relative error with v_2 is $\sqrt{1.5\times10^{-11}}/(6.0\times10^{-8}\sqrt{n}) \approx 645/\sqrt{n}$, so we need $n \approx (64500/x)^2$ to get x% relative error.

Another observation is that α (used for v_1) increases, and apparently converges to 1, when $\varepsilon \to 0$ for fixed n_i, or when n_i increases while ε is fixed. The intuitive explanation is that for this example, the dominant (most probable) paths leading to $\mathscr{F}$ are the direct ones, that involve failures of components of only one type, and these paths are all included in the computation of v_1.

CONCLUSION

Zero-variance simulation is an utopian ideal that can be achieved only in very simple situations where the quantities of interest can be computed exactly without doing any simulation. However, it can be approximated to a reasonable extent in several interesting situations, and can provide very large variance reduction factors in a practical way. Our numerical examples in this paper illustrate this: our proposed approach yields a low-variance estimator in a setting where all other previously proposed algorithms break their teeth. The method relies on a reasonable approximation of the function μ and this would generally depend on the model. Further studies with specific classes of models in different application areas are needed and should confirm the huge under-explored potential of this approach.

ACKNOWLEDGMENTS

This research has been supported by Grant OGP-0110050 and a Canada Research Chair to the first author, as well as EuroFGI Network of Excellence and INRIA's cooperative research initiative RARE to the second author. The paper was written while the first author was at IRISA, in Rennes, France.

REFERENCES

Ahamed I.; Borkar V.S.; and Juneja S., 2006. *Adaptive Importance Sampling for Markov Chains using Stochastic Approximation. Operations Research*, 54, no. 3, 489–504.

Alexopoulos C. and Shultes B.C., 2001. *Estimating Reliability Measures for Highly-Dependable Markov Systems, Using Balanced Likelihood Ratios. IEEE Transactions on reliability*, 50, no. 3, 265–280.

Booth T.E., 1982. *Automatic Importance Estimation in Forward Monte Carlo Calculations. Transactions of the American Nuclear Society*, 41, 308–309.

Booth T.E., 1987. *Generalized Zero-Variance Solutions and Intelligent Random Numbers*. In *Proceedings of the 1987*

Winter Simulation Conference. IEEE Press, 445–451.

Bucklew J.A., 2004. *Introduction to Rare Event Simulation*. Springer-Verlag, New York.

Cancela H.; Rubino G.; and Tuffin B., 2002. *MTTF Estimation by Monte Carlo Methods Using Markov Models. Monte Carlo Methods and Applications*, 8, no. 4, 312–341.

Fishman G.S., 1996. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Series in Operations Research. Springer-Verlag, New York.

Glynn P.W. and Iglehart D.L., 1989. *Importance Sampling for Stochastic Simulations. Management Science*, 35, 1367–1392.

Glynn P.W. and Whitt W., 1989. *Indirect Estimation Via* $L = \lambda w$. *Operations Research*, 37, 82–103.

Heidelberger P., 1995. *Fast Simulation of Rare Events in Queueing and Reliability Models. ACM Transactions on Modeling and Computer Simulation*, 5, no. 1, 43–85.

Juneja S. and Shahabuddin P., 2006. *Rare Event Simulation Techniques: An Introduction and Recent Advances*. In S.G. Henderson and B.L. Nelson (Eds.), *Simulation*, Elsevier, Amsterdam, The Netherlands, Handbooks in Operations Research and Management Science. 291–350. Chapter 11.

Kollman C.; Baggerly K.; Cox D.; and Picard R., 1999. *Adaptive importance sampling on discrete Markov chains. Annals of Applied Probability*, 9, no. 2, 391–412.

Law A.M., 2007. *Simulation Modeling and Analysis*. McGraw-Hill, New York, fourth ed.

L'Ecuyer P. and Tuffin B., 2006. *Splitting with Weight Windows to Control the Likelihood Ratio in Importance Sampling*. In *Proceedings of ValueTools 2006: International Conference on Performance Evaluation Methodologies and Tools*. ACM Publications, Pisa, Italy, 7 pages.

Nakayama M.K., 1996. *General Conditions for Bounded Relative Error in Simulations of Highly Reliable Markovian Systems. Advances in Applied Probability*, 28, 687–727.

Nakayama M.K. and Shahabuddin P., 2004. *Quick Simulation Methods for Estimating the Unreliability of Regenerative Models of Large Highly Reliable Systems. Probability in the Engineering and Information Sciences*, 18, 339–368.

Shahabuddin P., 1994a. *Fast Transient Simulation of Markovian Models of Highly Dependable Systems. Performance Evaluation*, 20, 267–286.

Shahabuddin P., 1994b. *Importance Sampling for the Simulation of Highly Reliable Markovian Systems. Management Science*, 40, no. 3, 333–352.

Tuffin B., 2004. *On Numerical Problems in Simulations of Highly Reliable Markovian Systems*. In *Proceedings of the 1st International Conference on Quantitative Evaluation of SysTems (QEST)*. IEEE CS Press, University of Twente, Enschede, The Netherlands, 156–164.

AUTHOR BIOGRAPHIES

PIERRE L'ECUYER is Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada. He holds the Canada Research Chair in Stochastic Simulation and Optimization. His main research interests are random number generation, quasi-Monte Carlo methods, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He is currently Associate/Area Editor for ACM Transactions on Modeling and Computer Simulation, ACM Transactions on Mathematical Software, Statistical Computing, International Transactions in Operational Research, The Open Applied Mathematics Journal, and Cryptography and Communications. He obtained the E. W. R. Steacie fellowship in 1995-97, a Killam fellowship in 2001-03, and he was elected INFORMS Fellow in 2006. His recent research articles are available on-line from his web page: `http://www.iro.umontreal.ca/~lecuyer`.

BRUNO TUFFIN received his PhD degree in applied mathematics from the University of Rennes 1 (France) in 1997. Since then, he has been with INRIA in Rennes. He spent 8 months as a postdoc at Duke University in 1999. His research interests include developing Monte Carlo and quasi-Monte Carlo simulation techniques for the performance evaluation of telecommunication systems, and developing new Internet-pricing schemes. He is currently Associate Editor for *INFORMS Journal on Computing*.

SIMULATION MODELS

Approaches of an Integrative Simulation Model for Project Development Processes

Sven Tackenberg
Bernhard Kausch
Morten Grandt
Christopher Schlick

Chair and Institute of Industrial Engineering and Ergonomics
RWTH Aachen University
D-52062, Aachen, Germany
E-mail: {S.Tackenberg, B.Kausch, M.Grandt, C.Schlick}@iaw.rwth-aachen.de

KEYWORDS
Decision Support, Forecasting, Discrete Simulation, Event oriented, Work Organization, Workflow Simulation

ABSTRACT

Development processes are difficult to plan successfully, and become more so with increases in complexity. Therefore, various types of methods and software tools are used by project managers to handle this complexity. In particular, a simulation capability enables a range of alternative scenarios for project development to be assessed at an early planning stage. The development of software solutions alone, however, is not sufficient. Instead, an extensive instrument consisting of methods, specification techniques and software tools for the integrated transformation and simulation of a graphical process model is needed.
This paper introduces C3 as a modeling method, especially designed to map weakly structured cooperative development processes. Additionally, a transparently structured transformation of a C3 model into two approaches of simulation models is demonstrated. Both approaches show how various stochastically generated project constellations can be compared using Petri-Net or object-oriented simulation.

1 INTRODUCTION

The increasing complexity of development projects, associated with shorter development times and an increasing cost pressure, raises the financial risks for enterprises. Nevertheless, sustainable competitive advantages under these basic conditions can be achieved if the management is able to implement an efficient work organizational scheduling of development processes. Thereby, the design of systems makes great demands on development methodology. There is a set of approaches that deal with the management of complex development projects. However, all these approaches are very generic and therefore the support to the project designer occurs in an abstract way (Redenius 2006). This is due to the high percentage of creative, heavily quantifiable activities as well as the many different influencing parameters. These are in particular the number of employees in a project, their individual qualifications and the available information and tools. Inefficient executions of projects are often the consequence of a missing consideration of influencing factors. A current study shows this insufficiency and indicates that approximately only 13% of work in projects is valuable (Gröger 2006).
Due to the plurality of the different influencing parameters, a development process must be considered as a sequence of activities in a three-dimensional room (Gausemeier et al. 2006). The arrangement of these activities in the room is affected by the characteristics of the influencing factors (Fig. 1). The following three dimensions span a project room:

- **List of Activities:** This list contains the activities necessary for the execution of a development project.
- **Project Time:** This dimension represents the time flow of the project from start to end. Furthermore, the project time illustrates the interrelation of synthesis and analysis, which leads from an abstract to a concrete problem solution.
- **Organizational Units:** These describe internal or external units of the enterprise (divisions, departments, etc.), which are authorized for the execution of activities.

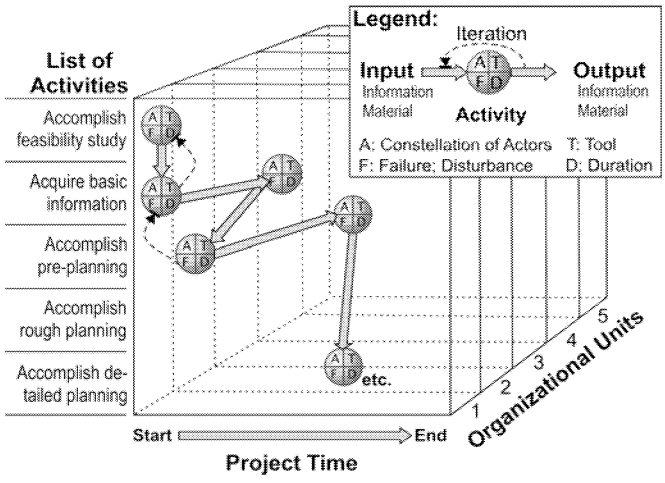


Figure 1: Project room of development processes

Each individual activity is determined by a multiplicity of individual factors. The four significant factors constellation of actors, necessary tools, estimated duration of activity and disturbance variables directly affect the project-specific progression inside the planning room. Therefore, a ubiquitous development process with a corresponding progression cannot exist (Frankenberger et al. 1998).
For the analyzing and benchmarking of distinctive development processes in the project room different approaches of simulation models have been developed at the Institute of Industrial Engineering and Ergonomics (Licht et

al. 2006, Schlick & Licht 2005). These models focus on the integration of factors that are critical to the success of a development project.
The approaches are based on the C3 modeling method, which was developed and validated in several development projects (Killich et al. 1999). All of them combine the advantages of participial modeling with the functionalities of a simulation environment.

2 SPECIFICATIONS OF COMPLEX DEVELOPMENT PROCESSES

The modeling and analysis of business processes and their corresponding structural organization by enterprises have continuously increased in the last years. An inquiry of German "IT decision makers" in the last ten years resulted in that four out of five enterprises are strongly or very strongly concerned with the model-supported analysis of business processes (Gadatsch 2005). Therefore, numerous methods for specifying a structural organization as well as a process-oriented organization of an enterprise are mentioned in scientific publications. Killich analyzed and evaluated several commonly used modeling methods and came to the result that none of them can satisfy the special needs of development processes (Killich et al. 1999). Existing models must particularly be enhanced with regard to the modeling of weakly structured, creative activities in domain-spanning world wide acting teams.
Based on the identified requirements, the C3 modeling method for weakly structured and domain-spanning development processes was developed (Killich et al. 1999). Its advantages consist of semiformal notation of the method, which permits a high degree of flexibility and creativity during the participative process modeling. Furthermore, the method allows a simple formalization and transformation of a C3 model into a simulation model.
Based on the Unified Modeling Language (UML) (Penker & Eriksson 2000), the C3 modeling language consists of 14 elements. Figure 2 shows the most important elements in a short section of a production system development model. The core elements for the characterization of an activity in the process room (Fig. 1), such as activities, resources and information, are also essential elements of a C3 model. These core elements are arranged in "swimlanes" according to their organizational affiliation. The core elements are supplemented by special elements which support the description of creative activities as well as activities with an afflicted uncertainty of execution. The so-called "blob" contains a quantity of activities without a pre-set sequence of execution. The connection between elements is specified by the following connectors:

- **Control flow:** Describes the connection between activities and the appropriated actors and tools.
- **Flow of information elements:** The flow of information elements describes the emitter, the receiver and the type of information.
- **Synchronous communication:** The characteristic of the synchronous communication is that all involved actors and resources must be available at the same time. If not, the synchronous communication cannot be executed.
- **Splitting node / Synchronization node:** The splitting note splits information. Therefore, a parallel execution of activities is possible. The synchronization node merges parallel activities.

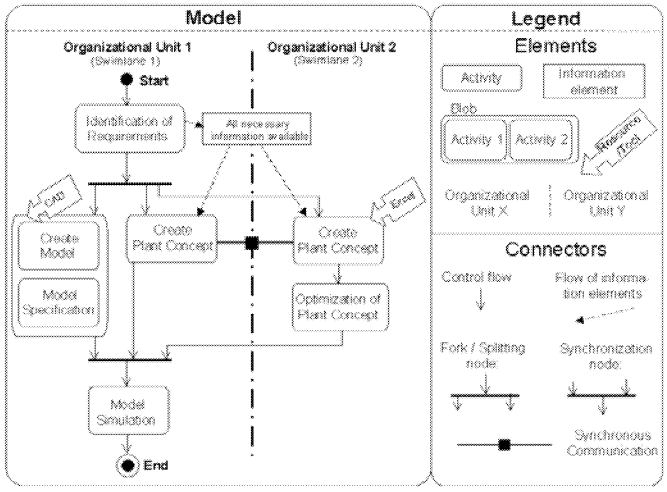


Figure 2: Core elements of C3 modeling language

3 BACKGROUND SIMULATION APPROACHES

In general, the progression of a development project is determined by the experience of the project team and its team leader. Therefore, software for supporting the work of project teams, such as MS-Visio or MS-Project, was launched to support the methods of business modeling (Tumay 1996). The static illustration of the progression of projects with Gantt charts, for example, visualizes the chronological progress and the appropriate application of resources. However, existing know-how from completed projects is often not sufficiently regarded in the planning of new development projects. One reason is the inadequate mapping and re-use of knowledge by business modeling methods and corresponding software solutions. One way to make implicit knowledge explicit can be achieved by a suitable knowledge management system. The second possibility consists of determining, by simulation, which project constellation is needed for a project.

Definition of Simulation Approach

A workflow simulation model of development projects in the chemical industry was developed at our institute in recent years. One way to differentiate simulation models is given by the level of detail found in human modeling. VDI-Guideline 3633 distinguishes between person-integrated models (person as reactive action model) and person-oriented models (display of various additional traits possessed by person) (Zülch 2004). Furthermore, simulation models of product development processes can, similar to VDI-Guideline 3633, be differentiated by two forms of model logic:

1. In the case of actor-oriented simulation models, system dynamics are produced by actors (modeled persons or organizational units) based on a specific task (Steidel 1994, Cohen 1992, Levitt et al. 1999).
2. In process-oriented simulation models system dynamics are produced by activities through the usage of resources

(persons, tools) (Browning et al. 2000, Cho et al 2005, Raupach 1999).

According to this definition, we present two concepts that are both a combination of a process-oriented simulation model and a person-oriented model.

4 INTEGRATIVE SIMULATION MODEL

The simulation model presented here offers a suitable technique for project planners in order to compare several alternative ways of project organization at an early stage. The simulation tool is embedded into comprehensive equipment for process modeling, process analysis and optimization. Therefore, we call it an integrative simulation model, which closes the existing gap between process modeling and an discrete-event simulation. This allows an integrated improvement of the process organization as well as the structural organization. From this objective the following requirements were derived:

- Process modeling should be done with the C3 modeling method and should be supported by software.
- The integrated transformation of a semi-formal C3 model into a formal simulation model should proceed automatically or interactively between software and planner.
- For the generation of different procedural-organizational alternatives of development projects and their subsequent simulation, the influencable factors relevant to success should be systematically varied.
- The dynamic visualization of the process progress should map the influence and the interrelations of the independent factors to the project progression. Therefore, the relevant influencable factors and their effects have to be identified.
- The layout of the graphical user interface should be based on human factors of software systems as well as experiences of project managers.

Based on the identified requirements, a software tool has been developed that allows the reusability and expandability of the simulation environment due to a component-oriented structure. This tool consists of the *C3-ProcessEditor*, the *SimulationTool*, the *Analyzer* and appropriate interfaces (Fig. 3).

The editor provides a tool for the graphical creation and parameterization of business processes based on the C3 method. It must be pointed out that the semi-formal C3 notation cannot map the entirety of factors relevant to simulation. As a result, the factors of influence, attributes and structures of the development process, which currently cannot be modeled with the C3 notation, have to be separately keyed in by the planner. The integration of the missing aspects through a modification of the C3 modeling method and a corresponding upgrade of the editor will be done in the current research project.

For the preset framework the SimulationTool generates several project constellations and project operations through the use of stochastic procedures. In the next chapter two developed simulation concepts will be introduced.

The Analyzer consists of a *BehaviorAnalyzer* and a *PerformanceAnalyzer*. While the first one dynamically illustrates the simulation run of a process progress, the PerformanceAnalyzer shows the dependence between independent and dependent variables by means of diagrams.

5 SIMULATION CONCEPTS

The success of a development project is up to the effective and efficient execution of activity sequences. Thus, the system dynamics within a project are determined through the sequence of activities during use of relevant resources. The first concept for simulation of complex development processes is based on time-extended higher level Petri-Nets. This concept was implemented and validated in numerous development projects within the process engineering industry (Kausch et al. 2005). The identified weak points of the Petri-Net based solution (cp. chapter 6) led to an extensive re-engineering of the approach as well as an alternative object-oriented simulation concept. While both

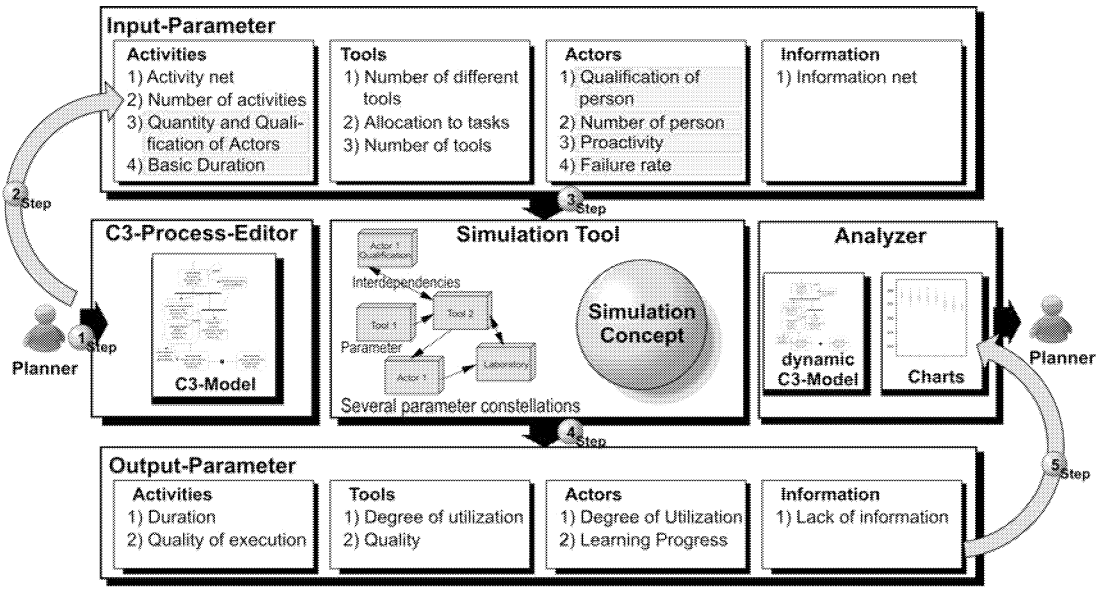


Figure 3: Structure of the integrative simulation model

approaches differ regarding implementation, the input parameters of the simulation and the generated parameters of the output are identical for both concepts (Fig. 3).
Figure 3 demonstrates the input parameters of the simulation model that can influence the output of the model. It is possible to differentiate between input and output parameters according to the described activities, tools, actors and information of the extended C3 model. The information regarding the successor and predecessor connections and the number of activities can be derived from the C3 model. Input parameters (grey deposited fields) that cannot be displayed, through C3 notation, are stored in matrices and are automatically imported to the model at the beginning of the simulation.

5.1 Simulation with time-extended higher level Petri-Net

For this approach the semi-formal C3 model, through use of transformation guidelines, is transformed into a simulation capable time-extended higher level Petri-Net. The transformation of the basic C3 elements activity, information and tools, along with their interaction relationships in a dynamic Petri-Net based simulation model, will be briefly introduced.

Transformation of Activities

In higher level time-extended Petri-Nets an activity represents a place that has a swimlane assigned to it according to the foundational C3 model. The sequence of activities and the structure of the process-oriented organization derived from this are transferred into the simulation model. The detailed specification of the places, such as information regarding the necessary actors, duration of an activity, etc., occurs through transitions. A transition that is forced between two places, or between the start and end points, verifies the fulfillment of structural and organizational requirements for the process flow.

Transformation of Interactions

The interactions between activities and resources are displayed in a Petri-Net through transitions. If the requirements of transitions are fulfilled the transition is switched on and the activity in its respective place can be carried out. The information flow is also regulated by requirements in a transition that verifies the existence of the necessary input information as a starting requirement for switching.

Transformation of Tools and Information Flow

The information and tools are displayed as places in the Petri-Net based simulation and are assigned to the appropriate swimlanes. The available tools and information describe input parameters of the simulation model. These are specified in the respective partial models through the hierarchical structure of higher level Petri-Nets, and are assigned to places in the simulation model.
The implementation of the Petri-Net based simulation concept was carried out with help of the java-based high-level Petri-Net simulator Renew, which was developed at the Department of Computer Science at the University of Hamburg (Kummer et al. 2006). The open-source code Petri-Net simulator Renew was chosen based on its graphical representation. A detailed evaluation of higher level Petri-Net simulators can be found in Störrle (1998). The structure of the simulation model encompasses the task network, four partial models, help networks and information lists.

Task Net

The complexity of distributed development projects demands transparent transformation of C3 models into a simulation-capable Petri-Net. The task network, as a key component of the simulation environment, depicts the process-oriented organization or activity relationships as a Petri-Net. Errors in the transformation process or during simulation can be easily identified by project planners through the high level of similarity between Petri-Nets and C3 notation. Figure 4 presents the Petri-Net which corresponds to the presented C3 model (Fig. 2).

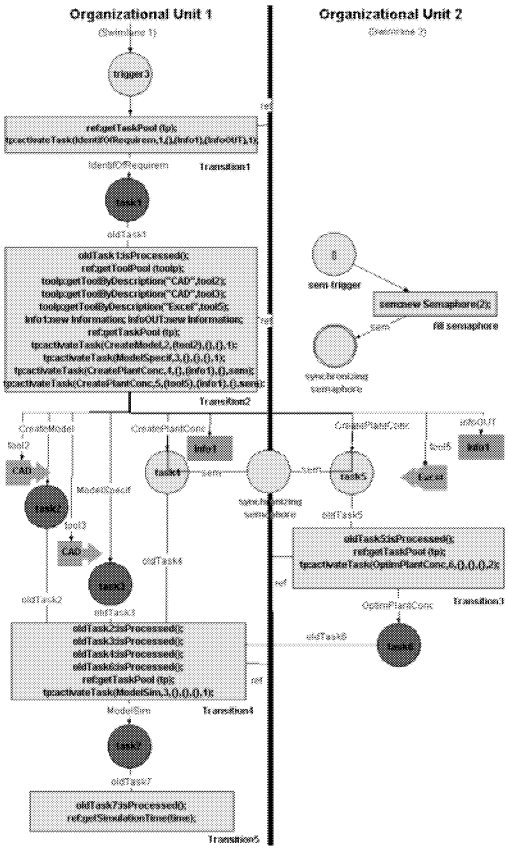


Figure 4: Structure of the Petri-Net based simulation model

Partial Models

The task network accesses disjoint partial models for the specification of the contained elements. These create a detailed representation of the behaviors of the individual process-oriented organizational elements. Pertaining to this are the partial models of activity, person, information and tools.
The *task network* consists of the tasks in the development process that need to be worked on. Within the tasks there is information about the subject matter needing to be processed, a necessary work tool, a profile of possible actors to do this processing, input and output information of the task as well as the expected duration needed to process the task. For the processing of a task, a qualified person and, if necessary, adequate tools are selected to achieve the goal. Each qualified person is assigned a value that reflects the occupational aptitude, dependent on the task at hand and the

required tools to complete the task. This value is calculated from the weighted sum of the person's assigned characteristics.
The *partial network of the actor* depicts the relationships between an actor (department, person, etc.) and activities. Personal qualifications and abilities are taken into account in the model concept in terms of recognizing that each person is able to act out a variety of activities. Significant attributes of the partial network person are:

- Proactivity of an employee: Each person is assigned a numerical value that describes the individual proactivity, i.e., output. This value improves the quality of the employee in the selection of the most qualified employee for a task (Kausch et al. 2007).
- Qualification in terms of a particular area of work: Skills are addressed to actors which qualify them for the processing of tasks in certain areas of work.
- Ability to deal with particular work tool: The actors possess abilities and qualifications that describe how well they can handle certain work tools. This means a actor must not only bear the appropriate qualifications to complete the task, but they must also have the ability to carry out the task through use of the necessary work tools.
- Learning aptitude: Due to routine tasks and new methods and expertise, certain qualifications can actually be improved. Alternatively, abilities not put to use over a greater period of time can also be weakened. This capacity to learn and unlearn is shown in a simulation model through a learning curve that is attributed to each person.

Information is represented through places in the Petri-Net in which a specific number of *partial models of the type information* exist. Information is assigned to tasks in the task network and has an influence on the duration of the development project.
The influence of work tools on the completion of tasks in the scope of the execution of activities through an employee is held in the *partial model of work tools*. The allocation of work tools to tasks results through the work organization of the development project. A tool can be used by only one employee at a time, though more than one tool can be used for a specific task. The amount of possible work tools cannot be exhaustively declared since the amount of possible tasks is in need of completion, detached from individual case examples, cannot be fully indicated.
The separation of task network and partial model ensures that the generation of various project scenarios – variation of the parameter configuration – is possible without modification of the task network. For the variation of the simulation parameters to describe different project scenarios, such as the number and characteristics of the actors and tools, the project planner is provided with a prototypically implemented interface. This interface integrates the available information currently in tables or databases to the simulation model and enables the modification of the data.

Help Networks
Help networks assist in the examination of simulation functionality and allow a modular program structure in the simulation environment Renew. Particularly significant are functionalities related to the selection of actors and tools for individual activities, the calculation of expected activity duration (basic duration) or the recording of simulation-relevant data.

Information Lists
The recorded simulation run data provides information about the start and end point of each activity, the total project duration, the processor of each activity, the tool usage and the learning progress of each actor. This information is graphically prepared for the planner through a user interface and is then presented (cp. chapter 6).

5.2 Object-oriented Simulation Model

The object-oriented C3*SimEngine* simulation model is designed with reusability and expandability in mind and consists of the *ControllingUnit*, *ActivityManager*, *Resource-Manager*, *DataParser* and *SimLogger*.

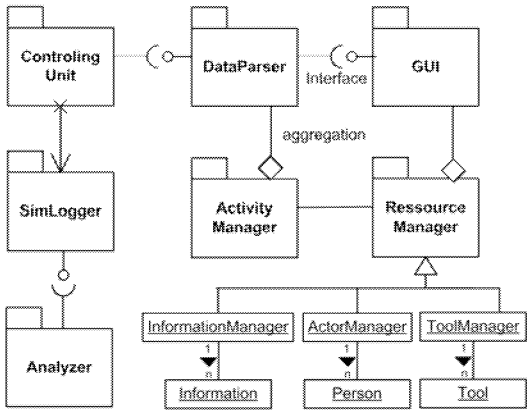


Figure 5: UML-based diagram

The **ControllingUnit** forms the core of the simulation environment and generates realistic combinations of activities and influencing factors as well as their characteristics – so-called simulation scenarios. In addition, simulation runs are directed and monitored from the ControllingUnit. Thus, the ActivityManager, Resource-Manager and SimLogger are called on depending on specific situations and they, for example, send information about the state of an activity or a tool back to the ControllingUnit.
The **DataParser** forms the interface of the simulation environment for the Graphical User Interface (GUI). The integration of a parser between the user interface and the simulation environment enables targeted selection and preparation of the input information. In this manner the future integration of process modeling tools, such as the above mentioned C3 editor or innovative AR solutions to participative process modeling, can be implemented without modification of the simulation routines.
The **ActivityManager** contains the entire activities within a development project. The ActivityManager receives an activity-based simulation scenario from the DataParser at the beginning of a simulation. For each activity of the C3 model this scenario contains a minimum of the following attributes:

- Activity ID and Name,
- Activity Status,
- Basic-Duration,

- Predecessor and Successor Activity and
- Necessary resources.

This list can be supplemented with additional attributes depending on the simulation's desired level of detail. The ActivityManager generates for each activity in the development process an object of the class *activity*.
The **ResourceManager** consists of an *Actor-*, *Tool-* and *InformationManager*. While the ActorManager and the ToolManager conduct the task-based and situation dependent allocation of actors and tools according to the planner's target criteria, the InformationManager generates and manages information dependent on the simulation or activity state. Parallel to the activities, an object of the respective class and with the relevant attributes is generated by the respective ResourceManager. Object-attributes of the class *actor* are, for example, job title, abilities and skills, proactivity, learning rate, etc. These actor-specific distinct attributes of the extended C3 model are prepared by the DataParser. Unoccupied attributes are automatically given a default value.
The **SimLogger** records the states of the simulation model during a run and produces the foundation for the visualization of results with the appropriate analysis tool. Simulation run experiences with the Petri-Net based simulation program Renew have shown that a dynamic visualization of the project flow (switching of transitions, etc.) is unnecessary for all generated scenarios since the planner is only interested in a small number of project constellations. In other words, the project configurations with the shortest, middle and longest project cycle durations are dealt with here. In order to attain better system performance the project cycles are dynamically visualized after the simulation run upon the planner's request. The user-friendliness of the simulation environment is significantly increased by a separation of the simulation and the graphical visualization and the resulting decrease in simulation time.
The object-oriented structure of the simulation environment makes the transformation of the C3 model into a simulation capable model easier. For each activity in the C3 model an object of the class *activity* is created with the corresponding attributes, such as name, predecessor activity, etc. All activities in the project are distinctly described according to the values of the individual attributes. Analogous to the transformation of the activities, this also occurs for the actors, tools and information. The process structure of a development project with this approach, unlike the Petri-Net based approach, is not described through a global task network in which the individual activities are carried out. For the approach followed here the individual object *activity* is assigned information about predecessor and successor activities and about the start, execution and end of an activity. The information about the process structure of a development process is thus stored in a decentral manner in the individual activities. The selected concept offers significant advantages regarding the future implementation of optimization procedures or the abstraction of project areas. Hence, with this approach the switching of activities in times of uncertainty, or the start of partial activities in the event of a particular state (e.g., an interruption or the completion of a sub task), can be simulated.

The object-oriented approach seizes the demands that result from a dynamically development process in a project room. The approach therefore offers the following three benefits:

- The modeled project and process structure shows a high level of flexibility regarding the configuration of various simulation scenarios due to the deployment of "autonomous" activity, information, tool and actor objects.
- Situation-specific decisions during a simulation run (e.g., occurrence of disruptions) and their impact on the project structure and the further project flow can be depicted via the parameterization and combination of the respective objects.
- The object-oriented approach, in contrast to the Petri-Net based approach, allows simple integration of optimization procedures in the simulation environment. Through object-oriented implementation in particular the procedures can be supplemented as desired and then applied, specific to a situation, to individual activities, tools, objects, etc.

6 VALIDATION – PETRI-NET BASED APPROACH

6.1 Experimental Design

The Petri-Net based simulation concept is validated by completed development projects of a manufacturing company and an industrial service provider from the chemical industry. The object-oriented approach are currently being implemented so that a timely evaluation of performance can occur.
In the test runs of the Petri-Net based simulation the number of persons was varied and afterward set to the optimal number. Following this, the number of the tools was also varied. The influence of these factors on the simulation time was then examined in order to judge the validity of the simulation model. Thus, process-oriented organization of the development projects was determined with the C3 method in several expert workshops. For this comparison the amount of persons was varied between two and eight and the quantity and type of available tools was varied between ten and 12. The distribution of expected processing duration (basic duration) was modeled as an additional independent variable. The basic duration for an activity is the starting part for the calculation of the probable duration of an activity. The calculated results follow a beta distribution.

6.2 Simulation Results

The results for a practical development process was first examined regarding significant differences in terms of the *Total Time of Work* – sum of processing times of all activities (TTW) and *Total Time of Project Duration* – duration between project start and end (TT_{PD}). To compare the central tendency of different groups of variables, the figure 6 offers a suitable overview.
The box-plots on the left show that the results of simulation are nearly beta distributed and no significant effect between *Total Time of Work* and *Number of Working Persons* can be assumed. The same result can be translated into the effect between TTW and *Total Number of Tools*. As can be seen on the right side of the figure 6, a clear qualitative effect for

the TT_{PD} between the configuration with two actors or a capacity of eight actors can be identified.

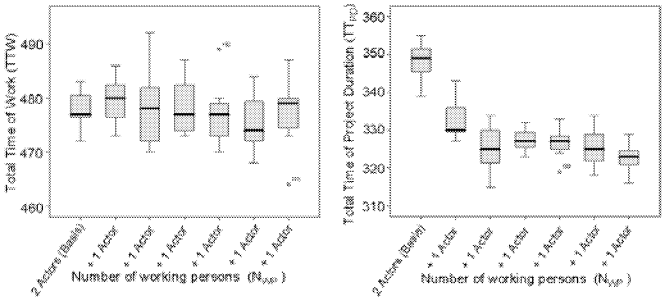


Figure 6: Simulation Results

The simulation further shows that the processing time can be reduced by approximately 8% through an assignment of more than four actors. However, no significant reduction in processing time can be measured following an assignment of more than four persons. This is due to the project structure's task network in which no more than four tasks can be simultaneously executed. Parallel processing by more than four workers can also not take place. These circumstances change, however, when several persons can simultaneously process one activity. This requires a division of labor within a task, the effects of which will be examined in future studies and which are not implemented in the current simulation concept.

To support the project manager a Gantt chart of the development project progression is created for every parameter constellation. Fig. 7 shows two Gantt charts for the shortest and longest project cycle time with constant resource allocation. The main effects for a shorter *Total Time of Project Duration* in the right chart occur through a greater quantity of simultaneously executed activities (Parallelitätsgrad – degree of parallelism), the thereby related more balanced capacity of the actors and the shorter *Total Time of Work*. The latter value indicates that the determined times are, on average, actually rather left-distributed within the right-skewed beta curve. This means that the underlying times are located in the left part of the curve.

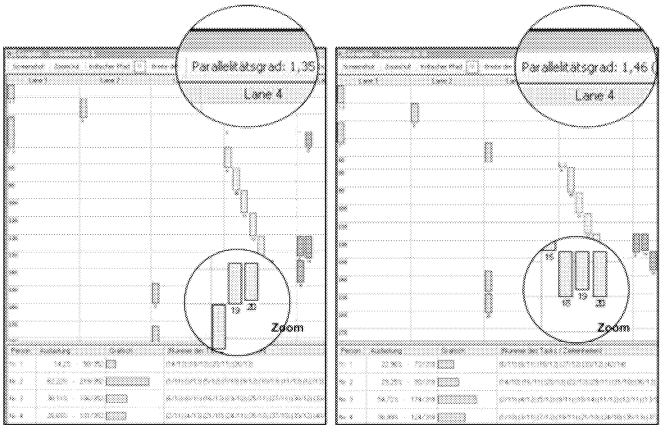


Figure 7: GUI for the task sequence

6.3 Evaluation

The Petri-Net based simulation approach offers a suitable technique for project planners in order to compare several alternative ways of project organization at an early stage, in respect to the number of involved persons or tools. The current assistance from the java-based higher level Petri-Net simulator Renew's implemented simulation approach points out the following weak points for complexly distributed development projects (80 activities, 20 different resource allocations):

- Subsequent changes in the C3 model lead to time-consuming adaptations to the Petri-Net created with Renew.
- The implemented software architecture – consisting of different sub-networks – is not very transparent and can thus be seen as purely a system for experts.
- System performance for complex development projects is suboptimal. The simulation of a simulation scenario (constant process organization, actor and tool configuration) with five runs (stochastic definition of processing duration and project flow) takes over an hour.
- An optimization of the project cycle according to defined target variables is not currently taken into consideration.

7 CONCLUSION AND OUTLOOK

7.1 Conclusion

Two new concepts for a simulation model based on the C3 modeling language were developed and offer project planners a suitable technique for quantitative comparisons of alternative ways of project organization at an early stage. Both concepts maintain the advantages of the C3 modeling method and additionally allow the transformation between this method and a Petri-Net based or an object-oriented simulation model. With these tools the influences of persons as well as tools for the total project duration or the total time of work can be easily investigated. The experiments with the Petri-Net based simulator produced satisfactory results as stated by the experts of different leading chemical engineering companies. The Petri-Net based simulation of complex and distributed development projects has shown that a large number of activities and different simulation scenarios causes a low performance of the simulation system. This can be explained through the implemented architecture as well as the performance of the Renew software. The presented object-oriented approach is currently implemented and will be validated by development projects of the process engineering industry.

7.2 Future Work

Both presented approaches – the Petri-Net based as well as the object-oriented simulation – will be further developed in the future in close collaboration with enterprises in the chemical engineering industry. Weak points of the Petri-Net based approach, e.g., the long simulation time per cycle for extensive development projects due to a low system performance or the laborious generation of simulation scenarios, will be solved by a comprehensive redesign. The preliminarily implemented object-oriented approach will be contemporarily validated in development projects of the process engineering industry. In the case of a successful validation, both simulation approaches will be tested and compared in a terminated process engineering development project.

In addition, attributes and correlations that were identified in the analyzed projects, and which are not implemented yet, will be added. These extensions regard, e.g., iterations, consideration of disturbance variables, and probability rich decisions or the formation of tool and task groups with similar job specifications. Furthermore, the correlations of individual factors are empirically calculated through the modeling of several example processes.

Acknowledgements
The research was funded by the German Research Foundation according to the transfer project TB61.

REFERENCES:

Browning, T.R., Eppinger, S. T.: Modelling the Impact of Process Architecture on Cost and Schedule Risk in Product Development. Massachusetts Institute of Technology, Sloan Scool of Management, Working Paper No. 4050, Cambridge, MA 2000

Cho, S.-H., Eppinger, D.: A Simulation-Based Process Model for Managing Complex Design Projects. In: IEEE Transactions on Engineering Management, 52, 3, P. 316–328, August 2005

Cohen, G.: The Virtual Design Team: An Object-Oriented Model of Information Sharing in Project Teams. Ph.D. Thesis, Stanford University, Palo Alto, CA, 1992

Frankenberger, E., Badke-Schaub, P., Birkhofder, H. (eds.): Designers – the Key to Successfull Product Development. Springer, Berlin et al, 1998

Gadatsch, A.: Grundkurs Geschäftsprozess-Management, Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. 4. Aufl., Vieweg, Wiesbaden, 2005

Gausemeier, J., Ebbesmeyer, P., Eckes, R.: Virtual Production - Computer Model-Based Planning and Analyzing of Manufacturing Systems, In: Dachtchenko, O. (Ed.) Reconfigurable Manufacturing Systems, Springer, Berlin, 2006

Gröger, M.: Wertschöpfungspotenzial Projektmanagement. In: REFA-Nachrichten 1/2006, Pp. 4-7, ISSN 0033-6874

Kausch, B., Schneider, N., Licht, T., Schmitz, P., Schmidt, L., Schlick, C.: Integrated Modeling and Simulation of Development Processes. In: The 2005 European Simulation and Modeling Conference, University of Porto, Portugal, Hrsg.: J. Manuel Feliz-Teixeira, J.; A.E. Carvalho Brito, EUROSIS-ETI, Ghent, Belgium, S. 283-290, 2005

Kausch, B., Grandt, M., Schlick, C.: Activity-based Optimization of Cooperative Development Processes in Chemical Engineering. In: SCSC 2007 "Summer Computer Simulation Conference", 15-18 July 2007, San Diego, 2007

Killich, S., Luczak, H., Schlick, C., Weissenbach, M., Wiedenmaier, S., Ziegler, J.: Task modelling for cooperative work. In: Behaviour & Information Technology, Hampshire, 18 5, S. 325-338, 1999

Kummer, O., Wienberg, F., Duvigneau, M.: Renew – the Reference Net Workshop. Release 2.1, May 2006. Available at: http://www.renew.de/

Levitt, R., Thomson, J. Christiansen, T., Kunz, J., Jin, Y. Nass, C.: Simulating Project Work Processes and Organizations: Toward a Micro-Contingency Theory of Organizational Design. Management Science, Informs 45:11; 1479-1495. 1999

Licht, T., Schlick, C., Dohmen, L., Schmidt, L., Luczak, H.: Improving multiple project management using simulation. In: Proceedings of the 4th Industrial Simulation Conference, 5.-7. June 2006, Palermo, Italy, Hrsg.: A. Genco, A. Gentile, S. Sorce, EUROSIS-ETI, Ghent, Belgium, S. 191-198, 2006

Penker, M., Eriksson, H.-E.: Business Modeling with UML. John Wiley and Sons, New York 2000

Raupach, H.-C.: Simulation von Produktentwicklungsprozessen. Dissertation, TU Berlin, Berlin, 1999

Redenius, A.: Verfahren zur Planung von Entwicklungsprozessen für fortgeschrittene mechatronische Systeme. Dissertation, Universität Paderborn, Paderborn, 2006

Schlick, C., Licht, T.: A Novel Approach to Simulate Complex Product Development Processes. In: Proceedings of the 12th International Conference on Industrial Engineering and Engineering Management (IE&EM '2005), 6.-8. Nov. 2005, Chongqing, China, S. 2-14 , 2005

Steidel, F.: Modellierung arbeitsteilig ausgeführter, rechner-unterstützter Konstruktionsarbeit – Möglichkeiten und Grenzen personenzentrierter Simulation. Dissertation, TU Berlin, Berlin. 1994

Störrle, H.: An Evaluation of High End Tools for Petri-Nets. In: Technischer Bericht, No. 9802, Ludwig-Maximilians-Universität München, 1998

Tumay, K.: Business Process Simulation. In: Proceedings of the 1996 Winter Simulation Conference, SCS, San Diego, CA, pp. 93-98, 1996

VDI-Richtlinie VDI 3633: Simulation von Logistik-, Materialfluss und. Produktionssystemen. Dez. 2001

Zülch, G., Jagdev, H., Stock, P., eds.: Integrating Human Aspects in Production Management. Springer, Berlin et al., 2004

SVEN TACKENBERG completed an apprenticeship as Electronician in the German electronics industry in 2000. He studied industrial engineering at the Technical University of Kaiserslautern, Oita University and received his degree from Paderborn University in 2006. From April 2006 to June 2007 he was a fellow of the International Graduate School Dynamic Intelligent Systems. Since June 2007 he is research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University of Technology.

BERNHARD KAUSCH studied engineering at the Technical University of Munich. His area of specialization was ergonomics and product development. Since August 2002 he is research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University of Technology.

MORTEN GRANDT received his M.S. degree (Dipl.-Ing.) in Safety Engineering from Wuppertal University in 1995, and the Ph.D. degree (Dr.-Ing.) also in Safety Engineering from Wuppertal University in 2004. From 1995 he worked as a scientific assistant and later on as provisional head of the Ergonomics and Information Systems department at the Research Institute for Communication, Information Processing and Ergonomics, Wachtberg, Germany. He is now head of the department Human-Machine-Systems at the Institute of Industrial Engineering and Ergonomics at Aachen University of Technology. He is chairman of the Ergonomics Chapter of the German Aerospace Society (DGLR) and member of NATO-RTO task groups.

CHRISTOPHER M. SCHLICK received his M.S. degree (Dipl.-Wirt.-Ing.) in Electrical Engineering from Berlin University of Technology in 1992, Ph.D. degree (Dr.-Ing.) in Mechanical Engineering from Aachen University of Technology in 1999, and the Habilitation degree (Dr.-Ing. habil.) also in Mechanical Engineering from RWTH Aachen University of Technology in 2004. He worked for the computer networks industry in 1992 and 1993 as a design engineer. From 1994 to 2000, he joined the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University of Technology. From 2000 to 2004 he was the head of the Department of Human-Machine-Systems at the Research Institute for Communication, Information Processing and Ergonomics, Wachtberg, Germany. He is now a full professor of Industrial Engineering and Ergonomics at Aachen University of Technology.

CONTRIBUTION TO THE DEVELOPMENT OF SIMULATION MODEL OF THE SHIP STEAM BOILER

Srđan Dvornik
Joško Dvornik
University of Split
Faculty of Maritime Studies
Zrinsko–Frankopanska 38, 21000 Split, Croatia
E -mail: sdvornik@pfst.hr

KEY WORDS

Marine steam boiler, simulation modelling, system dynamics, continuous and discrete simulation.

ABSTRACT

The aim of this paper is to demonstrate the successful application of system dynamic simulation modelling (System Dynamics Modelling – Jay Forrester – MIT) at investigating performance dynamics of the marine steam boiler.
Marine steam boiler is a complex non-linear system which needs to be systematically investigated as a unit consisting of a number of subsystems and elements, which are linked by cause-effect (UPV) feedback loops (KPD), both within the system and with the relevant surrounding.
In this paper the authors will present the efficient application of scientific methods for the research of complex dynamic systems called qualitative and quantitative simulation System dynamics methodology, which will allow for production and use of higher number and kinds of simulation models of the observed elements, and finally allow for the continuous computer simulation, which will significantly contribute to acquisition of new information about the non-linear character of performance dynamics of marine steam boilers in the process of designing and education.

SIMULATION MODELLING OF THE MARINE STEAM BOILER

Mathematical model of marine steam boiler

The boiler may be considered as a homogenous device, a thermal accumulator, i.e. a homogenous thermal capacity.

The results presented in the paper have been derived from the scientific research project „New Technologies in Diagnosis and Control of Marine Propulsion Systems“ supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

The equations of thermal balance of such thermal accumulator (capacity) suggest determining the equation for the level of the water in the boiler.
System dynamics mathematical model of the marine steam boiler is defined by explicit form of differential equations:

1. Equation of the boiler dynamics for the steam pressure:

$$\frac{d\varphi_K}{dt} = \frac{1}{T_{a1}} \cdot (\mu_G + a_1 \cdot \mu_V - a_2 \cdot \mu_P - a_3 \cdot \frac{d\mu_P}{dt} - k \cdot \varphi_K) \quad (1)$$

2. Equation of the boiler dynamics for the water level

$$\frac{d\varphi_Y}{dt} = \frac{1}{T_{a2}} \cdot (\mu_V - b_1 \cdot \varphi_K - b_2 \cdot \frac{d\varphi_K}{dt} - b_3 \cdot \mu_P - b_4 \cdot \frac{d\mu_P}{dt}) \quad (2)$$

Where the following denote:

φ_K - relative state of the steam pressure in the boiler,

φ_Y - relative state of the water level in the boiler,

T_{a1} - time constant of the steam boiler for the steam pressure [s],

T_{a2} - time constant of the steam boiler for the water level [s],

k - coefficient of self-regulation of the steam boiler,

μ_G -relative change of the position of the fue valve,

μ_V - relative change of the valve of the feed water,

μ_P - relative change of the position of the steam discharge valve,

$\frac{d\varphi_K}{dt}$ - speed of the change of the relative increment of the boiler steam pressure,

$\frac{d\mu_P}{dt}$ - speed of the relative change of the position of the steam discharge valve,

$a_{1,2,3}$ - coefficients of the steam boiler for the steam pressure,

$b_{1,2,3,4}$ - coefficients of the steam boiler for the water level.

System dynamics flowchart of the ship steam boiler

On the basis of the produced mental - verbal and structural models (*International Scientific Conference on Production Engineering*) the flowchart of the ship steam boiler in POWERSIM simulation language is produced at fig 1.

INVESTIGATING PERFOMANCE DYNAMICS OF THE MARINE STEAM BOILER IN LOAD CONDITIONS

After system dynamics qualitative and quantitative simulation models have been produced, in one of the simulation packages, most frequently DYNAMO or POWERSIM, all possible operating modes of the system will be simulated in a laboratory.
After the engineer, designer or a student have conducted a sufficient number of experiments, or scenarios, and an insight has been obtained about the performance dynamics of the system using the method of heuristic optimisation, optimisation of any parameters in the system may be performed, provided that the model is valid.
In the presented scenario the simulation model of the marine steam boiler for the steam pressure and the water level with two built-in PID governors will be presented.

1. Consumption of the steam is determined by the impulse function of 50-sec duration, which means from 200 – 250 sec, and MIP = 0, FIK = 0, FIY = 0.9999 at the initial TIME = 0.
2. Fuel supply MIG is determined as an outlet of PID-governor, at which inlet there is the discrepancy of the steam pressure (1-FIK) and correspondingly, the water supply MIV is outlet of the other PID-governor, to which the inlet is discrepancy (1- FIY).
3. Other parameters of the marine steam boiler equal nominal values.

Graphic results of the simulation:

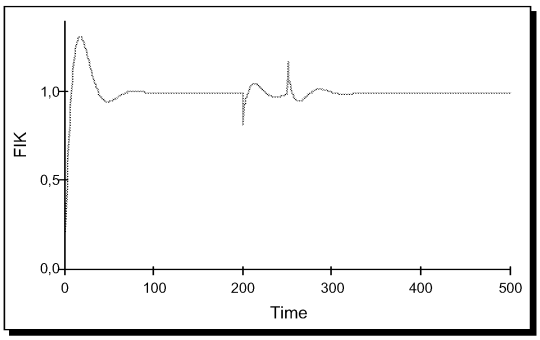


Figure 2. Relative state of the steam pressure in the steam boiler

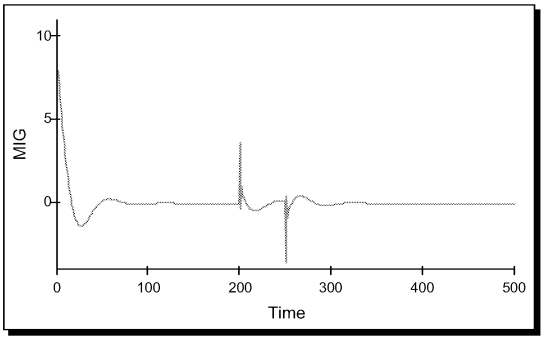


Figure 3. Relative change of the position of the fuel valve

From the results of the simulation it may be observed that the model shows real performance dynamics and that by applying PID governors and adequate values of coefficients better levelling and attenuation of the transition occurrence of the variables FIK and FIY will be achieved.

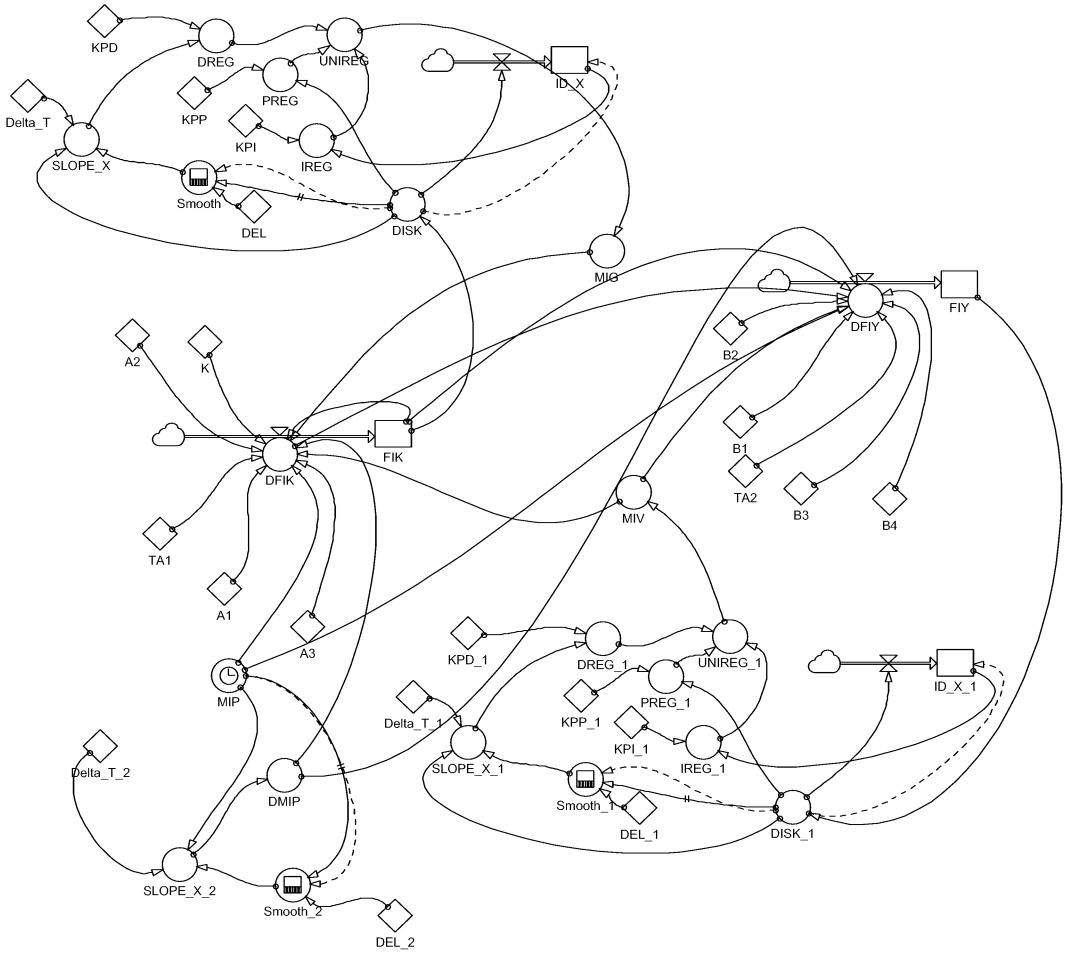


Figure 1. Global flowchart of the ship steam boiler with the built-in PID governors

CONCLUSION

System dynamics is a scientific method which allows simulation of the most complex systems. The method used in the presented example demonstrates a high quality of simulations of complex dynamic systems, and provides an opportunity to all interested students or engineers to apply the same method for modelling, optimising and simulating any scenario of the existing elements.

Furthermore, the users of this method of simulating continuous models in digital computers have an opportunity to acquire new information in dynamic systems performance. The method is also important because it does not only refer to computer modelling, but also clearly determines mental, structural and mathematical modelling of the elements of the system.

REFERENCES

Forrester, J.W. 1973/1971. *Principles of Systems*, MIT Press, Cambridge Massachusetts, USA.

Richardson, G.; and L. Aleksander. 1981. *Introduction to System Dynamics Modelling with Dynamo*, MIT Press, Cambridge, Massachusetts, USA.

Munitic, A. 1989. *Computer Simulation with Help of System Dynamics*, Croatia, BIS Split, p. 297.

Byrknes, A.H. *Run-Time User's Guide and Reference Manual*, Powersim 2.5, Powersim Corporation, Powersim AS, 12007 Sunrise Valley Drive, Reston Virginia 22091 USA.

Milic, L.; J. Dvornik.; and S. Dvornik. 2006. "Simulation of dynamic conditions of the ship steam boiler", *International Scientific Conference on Production Engineering*, 2006. Lumbarda, Korcula, Croatia, 137-143

Isakov, L.I.;and L.I. Kutljin. 1984. *Kompleksnaja avtomatizacija sudovljlh dizeljnih i gazoturbinmljih ustanovok*, Leningrad, Sudostreonnie.

Nalepin, R.A.; and O.P. Demeenko. 1975. *Avtomatizacija sudovljih energetskih ustanovok,*Leningrad,Sudostroennie.

Circles Model for Metro Light Rail Analysis

J. Manuel Feliz-Teixeira

Physics, Modelling and Simulation:

email: feliz@fe.up.pt
url: http://geinsrv.fe.up.pt/feliz

KEYWORDS: metro network, dynamic modelling, simulation and analysis, data projection, abstract domain, public transportation.

ABSTRACT

The present article shows a way of transforming a Metro Light Rail simulation model into a more abstract model, based on circles. This sort of representation, which goes into the abstract domain, is made to run in parallel with the common dynamic representation of the rail network during the simulation process, and gives the user precious information concerning the harmony with which the overall system is performing. Frequently, a simple look into this running *circles model* is enough to let the user identify or understand certain traffic effects associated with the accuracy of the vehicle scheduling, for example. Although abstract, this is an interesting representation to help study the dynamics of Metro Light Rail networks, together with other methods like sensitive analysis or the common train schedule or time-space diagrams. We also present some results obtained by simulating a didactic representation of the Metro Light Rail of Brussels, since it is a well-known and modern European Union (EU) light rail facility.

1. Introduction

Since the last decade the boom in urban traffic was slowly transforming into upsetting levels of congestion, stress, daily delays introduced in the movement of millions of people, and ultimately to unbearable levels of air pollution, one can notice a steady increase in the attraction of the Metro Light Rail as one of the most interesting transport solutions for citizen's flow within and between cities. We can now observe this trend spreading to several places in the world, following the good practice of countries that for a long time already have been prone to using rail transport, such as Germany and France, for example. It is therefore an excellent sign to know that even the emblematic town of Jerusalem is developing its first Metro Light Rail system by now.

It was more or less predictable that with time one should seriously opt for these fairly good quality transport networks in order to reduce not only the environmental impact of people's flow, but also in order to maintain the minimum acceptable quality of life in towns. Too many places in the world are already being confronted with high levels of stress and air pollution, obvious symptoms of a degradation of the life that people live. The price to pay for the compact urban traffic will, therefore, not only be measured in terms of pollution itself, but also, and very importantly, by the health level of the population, since this is the true energy of our towns and cities and the power of our production systems and industry. To replace the myth of comfort of our automobiles for the Metro Light Rail seems an excellent idea, in our opinion.

We have therefore decided to develop a general Metro Light Rail modeller/simulator based on the C++ core code for Supply Chain simulation owned by Feliz-Teixeira (2006), and to upon this create the appropriate interface and the metrics needed to represent and study these types of transportation systems. With such software, several models of real

Metro networks could already be created and tested by simulation[1], making experiments and comparing different policies; for instance, of injecting vehicles in the system. We could also measure some indexes of performance, as well as detect and analyse several unexpected network particularities, like the effect of the operation of terminals or control-lights in the formation of traffic queues, among other effects. All this is contributive to a superior understanding of the entire dynamics related to Metro Light Rail systems, and also to help finding some new ideas about how these systems should be operated. The future inclusion of demand indexes in this simulator is recently being considered, as well.

2. Metro Rail simulation

Although Metro Rail simulation is not a matter frequently seen at academic simulation conferences, perhaps because rail problems and its knowledge seem to be relatively stabilised, several software producers are focused on this subject for industrial purposes, training of rail controllers or of vehicle operators, and even for entertainment. *OpenTrack*, for example, is one of the most attractive Object-Oriented tools belonging to the first group, which is dedicated to modelling and simulating complex rail systems for the industrial level. It is quite user-friendly and was created with the objective of answering questions about rail operations, as mentioned by by Huerlimann (2006). Still on relatively the same level but perhaps less sophisticated, one could also point out *RailSim*, for example, from *Systra* (2007), or *RailModeller*, by MacRailSoft for those fond of the Mac Operating System, among others. Deserving reference is also *RailSys*, from Rail Management Consultants GmbH (RMCon), which is part of a more general software system for rail and metro network analysis and implementation (RMCon, 2000?). This tool has extensively been used in projects of Deutsche Bahn, Germany, the European Community, and the Network Rail of the United Kingdom, for example.

Beyond the scope of this article are those systems representing *human-on-loop* (Balci, 2003) simulations or games, as in the case of those tools dedicated to training operators or to entertain, for instance *PC-Rail*, *Microsoft Train Simulator*, and several others. These tools use very detailed 3D visualizations of the vehicles and the networks, and events are made to happen in the simulation in order to induce the operator to react and learn, a common practice found in flight pilot schools.

3. The *MetroModSim.exe* approach

Although one could say that our approach (see an overview in figure 1) is essentially on the same level as *OpenTrack*, it is being designed to address subjects more related to the tactical and the strategic spheres, instead of simply the operational. *MetroModSim.exe* may also be used to model and analyse some hard operational issues, like the particular operation of a terminus, for example, but at the moment, and due to the fact that our recent work is more directed at the *administrator* of the network instead of its *operator*, this approach seems the appropriate choice.

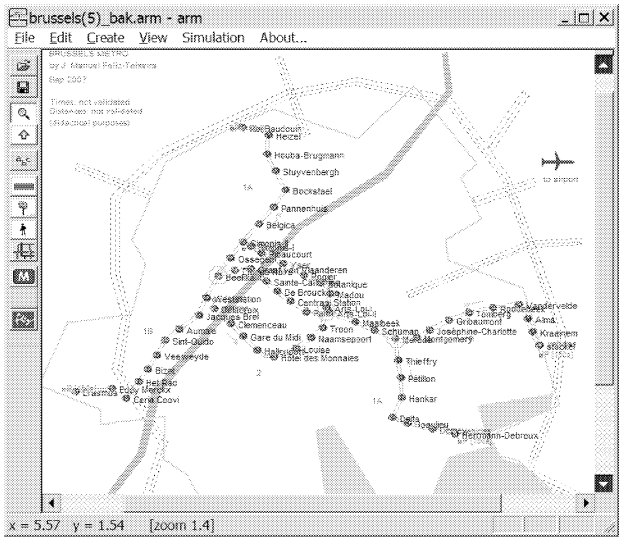


Fig. 1 An overview of *MetroModSim.exe*, showing a representation of the Metro Light Rail of Brussels.

Besides, in this approach we prefer to focus more on the *limits* of the operational, and not in particular on the operational itself, in order to avoid a direct interference with the engineering of the company who operates on the ground (*operator*), who is expected to own the hard operational information about the network and also the real experts on the matter. We therefore may establish the needs, as well as analyse the impact of certain operational policies in the dynamics of the overall

[1] Due to confidential reasons, we may not present here results obtained by simulating a real Metro Light Rail system recently studied by the author, since it is part of his consultancy activity for a private company dedicated to transport systems.

network, but we will always tend to look at the system from an upper perspective, detecting and evaluating its ultimate performances. How is the customer being served; how many vehicles could probably be saved; how many drivers; estimation of the timetables at each of the network nodes; evaluation of the capacities offered; visualization of traffic effects and detecting unexpected delays: all this can successfully be achieved by using *MetroModSim.exe*. We expect to publish more about *MetroModSim.exe* in articles to come.

The *circles mod*el, anyhow, must be considered a small component of this more general simulation tool, as well as a novel approach for representing *metro-lines* dynamics.

4. The Circles model

Usually, people who employ visual simulation tools have a tendency for representing systems as some sort of scaled models of reality, expecting with this to achieve better conclusions and carry out more perfect analyses. When systems get complex, however, such a mimic representation at the model level may induce in the analyst the same kind of complexity that already impregnates the reality, therefore making him/her somehow blind to the essential. Figure 2, for example, represents the Metro Light Rail of Brussels while being simulated in the standard window of *MetroModSim.exe.*

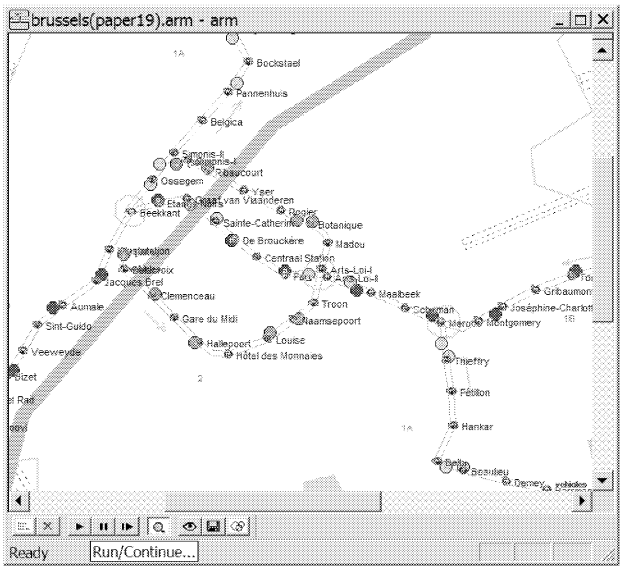


Fig. 2 The standard visual model, while running.

This is an example of the conventional visual representation of a detailed model running in a simulation window. The network is represented as a net of *arcs* and *nodes* which tend to mimic even the complexity of the space, and the vehicles are coloured circles running along this complex net. In the present case, the vehicles have been distributed by the three *metro-lines* of Brussels Metro[2]: *yellow 1A*, which connects Herrmann-Debroux to Roi-Baudouin; *red 1B*, connecting Stockel to Erasmus; and *orange 2*, around the city centre, linking Delacroix to Simonis.

When the simulation is running, the complexity emerging from this network becomes evident to the eyes of the observer, as one may easily understood from inspecting figure 2 again. In reality, since this kind of *view* mimics also the horizontal geography of the terrain, a clear relation between the diverse vehicles operating in the various *metro-lines* is not easily achievable, since it is obscured by the complexity of the terrain. This is, without any doubt, an excellent representation for detecting bottleneck effects, for example, and in many cases for understanding the behaviour of a certain light-control at a node, for instance, but it fails to give the analyst a more global view on the dynamics of the system, that is, an impression of the global balance/imbalance ratio of the operations. A judicious manager should have access to tools giving him/her such a global perspective on the operations space, since managing the system's parts is obviously not the same as having a wider view and a good understanding of the entire system.

The *circles model* is a method for avoiding such a tendency and to search for more abstract behaviours, or the "laws" which rule the system's complexity. We may consider that this is *projecting* reality onto the *abstract domain*, the domain of the "laws", or, if we are allowed to be a bit more dramatic, the "God's view", free of any superfluous geographic constrains.

So, if we exercise our minds in order to imagine that in effect a Metro network can be seen as a group of different *metro-lines* which may probably interact with each other at certain common points, or nodes, and if we also look at each of these

[2] This model was built with data collected from the Internet webpage of the Brussels Metro Light Rail, but the times and the distances are not real, since they were not crucial for the didactic purpose of this article. Therefore, they have been simply and roughly estimated.

metro-lines as a circle, which starts at *terminus#1*, runs to *terminus#2*, and again returns to *terminus#1* in order to reinitiate a new cycle of transport; instead of representing the network as a scaled version of geographical paths and points, as previously, we may represent it as a group of running circles, as shown in the next figure.

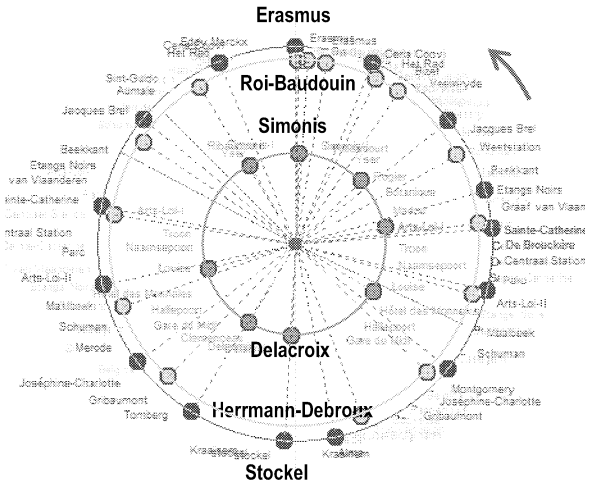


Fig. 3 The Metro Light Rail of Brussels, represented as a *model of circles*, using *MetroModSim.exe* (3 metro-lines).

These circles, each of which stands for a *metro-line*, with a perimeter directly proportional to the distance of going and returning to the main terminus (*terminus#1*), will sustain the vehicles moving during the simulation in each of the corresponding *metro-lines*. It is therefore extremely easy not only to understand how the overall system is performing, but also to detect any asymmetry induced by some sort of event or procedure, like a deficient operation at a terminus, for example. In effect, one can easily perceive from figure 3 that all *metro-lines* seem relatively well balanced, with their vehicles similarly spaced between each other, which means good stability in the frequency offered to the customers; even if there will probably be some agglomeration in the "*yellow 1A*" line in the access to Roi-Baudouin. The "*orange 2*" line, however, operating with only 8 vehicles, appears slightly imbalanced, since there are some "empty" spaces between certain vehicles. Due to this fact, we may expect less stability in the frequency offered in this line. Notice that the number of vehicles injected in this model has been estimated for the purpose of this article, so it does not probably express the real conditions of operations on the ground. A serious analysis would, of course, require collecting and validating the appropriate data.

Mathematically, however, there are also some interesting results emerging form this *circles model* point of view, like the calculation of the *optimum number of vehicles* that should be injected into a certain *metro-line*, the calculation of the frequency change effect when changing the length of a *metro-line*, how balanced the network is performing, etc., but we expect to treat these mathematical aspects in some articles to come. For the moment, we would simply present some more considerations about the "*orange 2*" *metro-line* of the Brussels network.

5. The *orange 2* imbalance, and its correction

As an exercise, we invite the reader to focus the attention on the "*orange 2*" line, which dispatches people around the city centre, as one can see from figure 4.

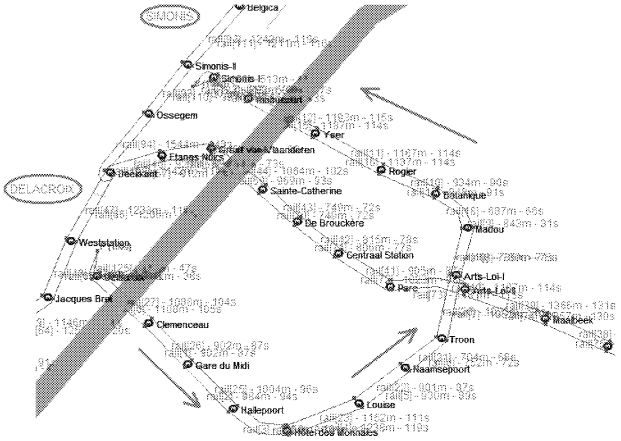


Fig. 4 The *orange 2* line, serving the centre of Brussels.

In order to better illustrate the idea of the *circles model*, and to make some experimentations by simulation, we have decided to model this *metro-line* with origin in Delacroix (*terminus#1*), where the vehicles have been injected, and with destiny Simonis (*terminus#2*), where the vehicles were to invert in order to start operating in the opposite direction. After some rough calculations we have chosen to inject 8 vehicles into this *metro-line*, separated by around 7,5 minutes, as a first bet. The simulator was then made to run for nearly 3 hours of operations, leading to the following frequency figure offered to the customer:

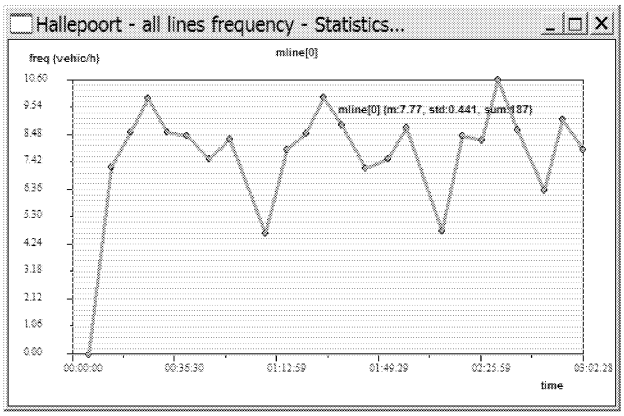


Fig. 5 Frequency offered to the customer (*orange 2*).

This means that an average of 7,8 vehicle/hour could be expected by the customers of this *metro-line*, with a deviation of ± 1,8 vehicle/hour (in a 4σ confidence interval).

But, if we now inject one more vehicle into this same *metro-line*, the resulting *circles model* already shows a better balanced situation (see Fig. 6).

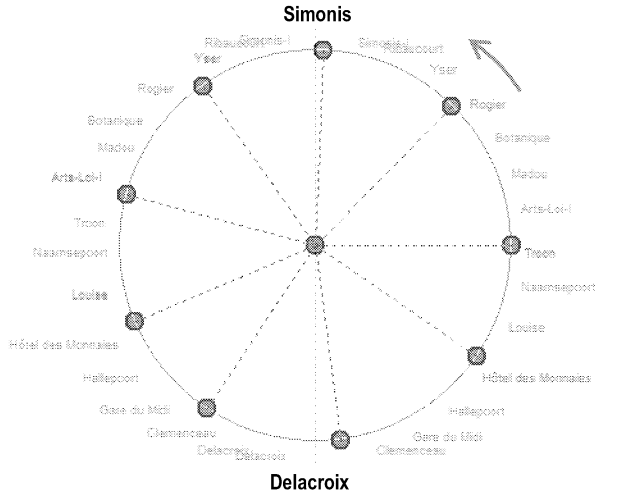


Fig. 6 Circles model of the *orange 2* line, operating with 9 vehicles separated by nearly 7,5 minutes.

At the same time, an obvious and slightl increase in the frequency offered to the clients was also observed, as shown in figure 7.

This simple examplemakes clear the potential available in the *circles model* method for analysing not only a metro network but also several others transport systems. We expect to soon also apply it to bus and rail simulation, for example, and probably it will even make sense to use it in airport and pedestrian studies.

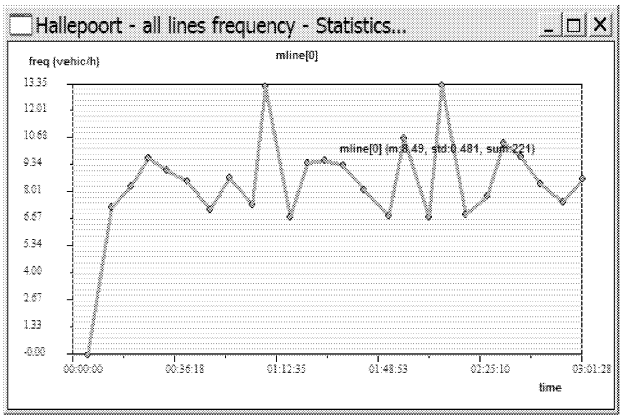


Fig. 7 Resulting increase in the frequency offered to the customer, in the *orange 2*) line.

We believe, however, that the perspective of *projecting the reality onto the abstract domain* is the most important idea contained in the *circles model*, since it may be an excellent way to try to understand the complexity of the systems we operate and that surround us.

6. Conclusions

The results obtained in several practical studies, and in particular with the didactic simulation of the Brussels Metro Light Rail presented in this article, are leading us to believe in the great utility of the *circles model* method. We consider this method a sort of *projection of the reality onto the abstract domain*, and an extremely useful representation to explore and understand the subtle "laws" which frequently impregnate dynamic systems, but can be obscured by an excessive tendency for representing the phenomena simply as they appear to our eyes. We advocate that a better and superior understanding of the complexity can be achieved by looking at the reality in the perspective of a more abstract mindset. In effect, as was shown in this article, a simple look into the running *circles model* of a *metro-line* is frequently enough to let us understand its main problems and identify certain ways to solve them, mostly based on the simpler idea of a *balanced/imbalanced* ratio.

Besides, the *circles model* view is also leading us to understand that a *metro-line*, like a bridge or a molecule, has a certain *proper frequency* for being operated, which may be seen as the frequency to which its imbalance is minimal. More than that will

represent too much pressure on the *metro-line*; less than that will mean too little pressure on it.

Finally, we would be pleased to receive any comments or feedback about any experiments based on these ideas.

Author Biography:

J. Manuel Feliz-Teixeira graduated in Physics in the Faculty of Sciences of University of Porto, Portugal, and received an MSc and PhD from the Faculty of Engineering of the same university. His work has been related to various matters, from optical communications, solar energy and seismology to, more recently, the simulation of complex systems in management science, like warehouse, supply chain, production systems and transport systems. His PhD thesis is on “Flexible Supply Chain Simulation”.

References:

Balci, O. (2003). *Verification, Validation, and Certification of Modeling and Simulation Applications.* Paper presented at the 2003 Winter Simulation Conference, USA.

Feliz-Teixeira, J. M. (2006). *Flexible Supply Chain Simulation (published thesis)*. Porto, Portugal: Publindustria Publisher.

Huerlimann, D. (2006). *OpenTrack: Simulation of Railway Networks*. OpenTrack. 2007

RMCon. (2000?). *RailSys, Timetable and infrastructure management*. Rail Management Consultants GmbH (RMCon). 2007, from http://www.rmcon-australia.com/uber_railsys.html

SYSTRA. (2007). *RailSim 8 network simulator*. SYSTRA Consulting, Inc., 2007, from http://www.railsim.com/NetSim.htm

AN EVOLUTION MODEL OF B-SPLINES PARAMETRIC SURFACE

Manuel Gonzàlez-hidalgo, Arnau Mir and Gabriel Nicolau-Bestard
Computer Graphics, Vision and Artifitial Intelligence Group.
Maths. and Computer Science Department.
University of the Balearic Islands. Spain
email: manuel.gonzalez@uib.es arnau.mir@uib.es gabriel.nicolau@gmail.com

KEYWORDS
B-splines, Finite Elements, Variational Formulation, Dynamic Deformation Model.

ABSTRACT

The study of the motion of deformable objects is one of the most important topics in computer animation, computer modelling, and so on. This paper presents an evolution model which deform a parametric surface with a linear complexity and nearly stability. The deformation model is based on an associated energy to one surface, that it check the shape of it. The associated variational formulation to the problem of minimize the energy functional is solved using the finite element method based on B-splines. The spatial discretization where these finite elements are defined and computed shows as a reduced number of control points is deformed instead of all the surface points, obtaining an efficient numerical scheme.

INTRODUCTION

The deformation models include a large number of applications, and they have been used in fields as edge detection, computer animation, geometric modelling, and so on. In this work, a deformation model will be introduced that uses B-splines as finite elements. The model includes deformation equation, numerical method to solve it, examples of deformations, computational cost and study of stability.
The use of B-splines and their properties as finite elements was introduced by Höllig in Höllig (2003). The model of deformation is based on the model used in Cohen (1992) where the finite elpements are classical finite elements (triangles and squares). We choose B-splines as finite elements to solve our deformation equation.
The use of classical finite elements makes a mosaic that it fills all space with a large computational cost if we want to solve our problem because a big data structure is needed. Otherwise, the B-splines finite elements combine the computational advantage of B-splines and standard mesh-based elements because the obtained data structure is smaller than the data obtained using classical finite elements.
This work can be considered an extended version of González-Hidalgo et al. (2006) presented in CompIMAGE'06 Conference and it's organized as follows: in section 2, we introduce B-splines finite elements and their properties; in section 3, the model of surface deformation, the numerical resolution and the computational cost are developed. In section 4, we show some illustrative examples. Section 5 is devoted to analize the stability of our numerical method; in sections 6 and 7, the influence of first and second order parameters are estudied. Finally, in section 8, some conclusions are exposed. More details about sections 2, 3 and 4 can be found in González-Hidalgo et al. (2006).

B-SPLINES

The B-splines are piecewise polynomial functions with a good local approximations for smooth functions Piegl (1997). We have choosed B-splines as piecewise polynomial approximation because of its local support. This property reduces the computational cost of the model. Uniform B-splines can be defined in several ways de Boor (1978) Farin (1997) Piegl (1997) Höllig (2003). In this work we have taken the definition given by Höllig in Höllig (2003), which it will be described next.

Definition 0.1 *Höllig (2003) An uniform B-spline of degree n, b^n, is defined by*

$$b^n(x) = \int_{x-1}^{x} b^{n-1}(t)dt$$

$$\textit{starting with } b^0(x) = \begin{cases} 1, & x \in [0,1), \\ 0, & \textit{otherwise.} \end{cases}$$

To evaluate the B-splines in a simple form and fast computationally, we use the following recurrence equation (De Boor de Boor (1978) and Cox Cox (1972)).

$$b^n(x) = \frac{x}{n} b^{n-1}(x) + \frac{(n+1-x)}{n} b^{n-1}(x-1) \qquad (1)$$

In order to construct the finite elements bases, we will use a scaled and translated uniform B-spline. They are defined by transforming the standart uniform B-spline, b^n, to the grid $h\mathbb{Z} = \{..., -2h, h, 0, h, 2h, ...\}$, where h is the scaled step.

Definition 0.2 *The transformation for $h > 0$ and $k \in \mathbb{Z}$ is $b^n_{k,h}(x) = b^n(\frac{x}{h} - k)$. The support of this function is $[k, k+n+1)h$*

The first order derivative of the transformed B-spline is given by

$$\frac{d}{dx} b^n_{k,h}(x) = h^{-1}(b^{n-1}_{k,h}(x) - b^{n-1}_{k+1,h}(x)) \tag{2}$$

Höllig (2003). Higher order derivatives can be computed as a linear combination of lower degree B-splines:

Theorem 0.1 *The m^{th} derivative of a degree n transformed B-spline following the definition 0.2 is given by the recurrence relation*

$$\frac{d^m}{dx^m} b^n_{k,h}(x) = h^{-m} \sum_{i=0}^{m} (-1)^i \begin{pmatrix} m \\ i \end{pmatrix} b^{n-m}_{k+i,h}(x) \tag{3}$$

Obviously, this equation has sense if $m \leq n$ since in others cases the derivative is 0.

The generalization of one dimensional B-splines is described in the following construction. The N-variate B-spline of degree $\mathbf{n} = (n_1, ..., n_N)$, of index $\mathbf{k} = (k_1, ..., k_N)$ and the space discretization $\mathbf{h} = (h_1, ..., h_N)$ is defined as

$$B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(\mathbf{x}) = \prod_{i=1}^{N} b^{n_i}_{k_i,h_i}(x_i). \tag{4}$$

The support of this function is $\prod_{i=1}^{N} [k_i, k_i + n_i + 1)h_i$. Applying basic properties of differential calculus and applying theorem 0.1, a compact expression for any partial derivative of multivariate B-spline can be obtained. Using theorem 0.1, the derivatives can be evaluated and they are obtained with a smaller computational cost, since less recurrences are applied.
A B-spline parametric surface is given by $S : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ where

$$S(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} P_{\mathbf{k}} B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(\mathbf{x}) \tag{5}$$

(see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007)) where $P_{\mathbf{k}}$ are the so called the *control points* and they are the elements that determine the B-spline surface.

DEFORMATION MODEL AND NUMERICAL RESOLUTION

The deformation model is based on an associated energy to one surface, that it checks the shape of it.
The energy function is a non convex function with a local minimum. The goal is to achieve this minimum using an evolution model. This mininum depends on the initial surface and the used evolution model.

The associated energy functional, $E : \Phi(S) \to \mathbb{R}$, $S \mapsto E(S)$, is defined as

$$E(S) = \int_\Omega \omega_{10} \left|\frac{\partial S}{\partial u}\right|^2 + \omega_{01} \left|\frac{\partial S}{\partial v}\right|^2 + \omega_{11} \left|\frac{\partial S}{\partial u \partial v}\right|^2 + \omega_{20} \left|\frac{\partial^2 S}{\partial u^2}\right|^2 + \omega_{02} \left|\frac{\partial^2 S}{\partial v^2}\right|^2 + \mathcal{P}(S(u,v)) du dv$$

Terzopoulos (1986), Cohen (1992), Montagnat et al. (2001), where $\mathcal{P}$ is a potential of the forces that works on the surface. Using the equations of Euler-Lagrange, it can be proved Cohen (1992) that an energy local minimum must satisfy:

$$-\omega_{10}\frac{\partial^2 S}{\partial u^2} - \omega_{01}\frac{\partial^2 S}{\partial v^2} + 2\omega_{11}\frac{\partial^4 S}{\partial u^2 \partial v^2} + \omega_{20}\frac{\partial^4 S}{\partial u^4} + \omega_{02}\frac{\partial^4 S}{\partial v^4} = -\nabla \mathcal{P}(S(u,v)) + \text{boundary conditions} \tag{6}$$

The surface domain is $\Omega = [0,1]^2$ and the boundary conditions are: $S(u,0) = (u,0,0)$, $S(u,1) = (u,1,0)$,$S(0,v) = (0,v,0)$, $S(1,v) = (1,v,0)$.

VARIATIONAL FORMULATION

With the purpose of establishing the variational formulation of the boundary value problem done by (6), we recall the definition of a Sobolev Space of order two $H^2(\Omega)$,

$$H^2(\Omega) = \{S \in L^2(\Omega) \,:\, \frac{\partial^{(\alpha_1+\alpha_2)} S}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}} \in L^2(\Omega), \; 0 \leq \alpha_1 + \alpha_2 \leq 2\,, \alpha_1, \alpha_2 \in \mathbb{N}\}$$

We will consider the set of functions $(H^2(\Omega))^3$ satisfying the previous boundary conditions. We will denote this set by $\mathcal{H}$. It can be proved Cohen (1992) Raviart (1992) that solving equation (6) is equivalent to find an element $S \in \mathcal{H}$, such that $a(S,T) = L(T)$ for all $T \in \mathcal{H}$, where $a(\cdot,\cdot)$ is a bilinear form defined as

$$a(S,T) = \int_\Omega \left(\omega_{10}\frac{\partial S}{\partial u}\frac{\partial T}{\partial u} + \omega_{01}\frac{\partial S}{\partial v}\frac{\partial T}{\partial v} + 2\omega_{11}\frac{\partial^2 S}{\partial u \partial v}\frac{\partial^2 T}{\partial u \partial v} + \omega_{20}\frac{\partial^2 S}{\partial u^2}\frac{\partial^2 T}{\partial u^2} + \omega_{02}\frac{\partial^2 S}{\partial v^2}\frac{\partial^2 T}{\partial v^2} \right) du dv \tag{7}$$

and $L(\cdot)$ is the following linear form

$$L(T) = -\int_\Omega \nabla \mathcal{P}(S) T du dv.$$

DISCRETIZATION

We want to find a function $S \in \mathcal{H}$ such that

$$a(S,T) = L(T), \; \forall T \in \mathcal{H}. \tag{8}$$

In order to do this, the surface domain will be discretized. The spatial discretitation is given by $h_1\mathbb{Z} \times h_2\mathbb{Z}$ where $h_1 = \frac{1}{N_1+n_x-1}$ and $h_2 = \frac{1}{N_2+n_y-1}$, where $N_1 \times N_2$ are the control points of the B-spline surface that we will to evolve (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007) for details). This spatial discretization will fix the control points that are not zero. The index $\mathbf{k} = (k_1, k_2)$ belongs to the set $\{-n_x, ..., N_1 + n_x - 1\} \times \{-n_y, ..., N_2 + n_y - 1\}$. So, the B-spline surface will come determined by the relevant B-splines, and they are specified by the following equation

$$S(u,v) = \sum_{k_1=-n_x}^{N_1+n_x-1} \sum_{k_2=-n_y}^{N_2+n_y-1} P_{(k_1,k_2)} B^{\mathbf{n}}_{(k_1,k_2)\mathbf{h}}(u,v). \quad (9)$$

The B-spline bases are determined by the following set of finite elements of finite dimension: $V^{\mathbf{n}}_{\mathbf{h}} =< \{(B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(u,v), 0, 0) : \mathbf{k} \in \mathcal{I}\} \cup \{(0, B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(u,v), 0) : \mathbf{k} \in \mathcal{I}\} \cup \{(0, 0, B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(u,v)) : \mathbf{k} \in \mathcal{I}\} >$ where $\mathcal{I} = \{0, \ldots, N_1 - 1\} \times \{0, \ldots, N_2 - 1\}$. Thus, taking into account the boundary conditions, the control points $P_{\mathbf{k}}$ associated to B-splines belonging to the set $V^{\mathbf{n}}_{\mathbf{h}}$ are the unique ones that are computed using the equations (10) and (11) (see below).
Using equations (8) and (9) we can obtain three linear systems, one for each coordinate:

$$AP_i = L_i, \quad i = 1, 2, 3, \quad (10)$$

where A is a square matrix and their elements are:

$$a((B^n_{k,h}, 0, 0), (B^n_{j,h}, 0, 0))_{(k,j)\in\mathcal{I}\times\mathcal{I}},$$

P_i is a vector of component i of each control point and L_i is a vector with components $L_1 = L((B^n_{k,h}, 0, 0))_{k\in\mathcal{I}}$, $L_2 = L((0, B^n_{k,h}, 0))_{k\in\mathcal{I}}$, $L_3 = L((0, 0, B^n_{k,h}))_{k\in\mathcal{I}}$.
The computational cost of A is constant and the computational cost of L_i, $i = 1, 2, 3$ is $O(N_1 \times N_2)$ (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007)).

Dynamic evolution model
The classical dynamical model of evolution has been applied Cohen (1992), Qin (1997), Montagnat et al. (2001), González et al. (2001), Mascaró (2002) Mascaro et al. (2002). In our dynamic model, the surface depends on time. So, we have $S(u, v, t)$. Nevertheless, this dependency only affects to the control points which is an advantage since in each iteration we do not have to calculate all the surface. Therefore, we only must calculate the new control points. Thus, our dynamic model of evolution comes determined by the equations

$$M\frac{d^2P_i}{dt^2} + C\frac{dP_i}{dt} + AP_i = L_i, \ i = 1, 2, 3. \quad (11)$$

where M and C are the mass and damping matrices respectively and they are diagonal matrices.
The dynamic system (11) has been discretized in time using the standard finite-difference approximation operators. A scheme based on central finite-differences has been used:

$$M\frac{P_i(t+\Delta t) - 2P_i(t) + P_i(t-\Delta t)}{\Delta t^2} + C\frac{P_i(t+\Delta t) - P_i(t-\Delta t)}{2\Delta t} + AP_i(t) = L_i, \ \ i = 1, 2, 3. \quad (12)$$

So, to solve equation of the scheme, we have to solve the following three linear system of equations for $P_i(t+\Delta t)$, associated to an explicit integration procedure for the ordinary differential equation:

$$G \cdot P_i(t+\Delta t) = H_i, \ \ i = 1, 2, 3, \quad (13)$$

where $G = 2M + \Delta t\, C$ and $H_i = 2\Delta t\, L_i - \left(2\Delta t^2\, A - 4M\right) P_i(t) - (2M - \Delta t\, C)\, P_i(t - \Delta t)$ (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007) for details).
The previous system is easy to solve because matrix G is diagonal. Then, the original nonlinear differential equation (11) has been reduced to a sequence of diagonal linear algebraic systems done by (13). And, taking in to account the set of finite elements that determines our B-spline bases, this algebraic system of equations is solved only for a reduced number of control points.
The used numerical scheme in the dynamic model depends on two previous iterations $P^{t-\Delta t}$ and P^t, taking the same P^0 and P^1, where $P^0 = S(u, v, 0)$.
This dynamic model requires $O(N_1 \times N_2)$ for each time step in order to solve the three linear systems (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007) for details)

EXAMPLES

This section shows several examples of deformations obtained applying our dynamical model. All the experiments displayed in this section has been made using $\omega_{10} = \omega_{01} = 0.1$ and $\omega_{11} = \omega_{20} = \omega_{02} = 0.01$ and $\Delta t = 0.1$.
In Figure 1, we show several iterations obtained using the dynamic model with bicubic B-splines, a force $(0, 100, 0)$ and $N_1 \times N_2 = 25$, The force is applied only at the central point of the surface.
Not only we can apply forces in vertical directions. Also, we can apply forces in other directions as we can see in Figure 2, where we have applied over all the surface domain an oblique force in the direction $(1, 4, 1)$, with module 106, taking bicubic B-splines and $N_1 \times N_2 = 49$
.
Figure 3 shows several iterations obtained using the vertical force $(0, 200, 0)$ over all the surface, using bicubic

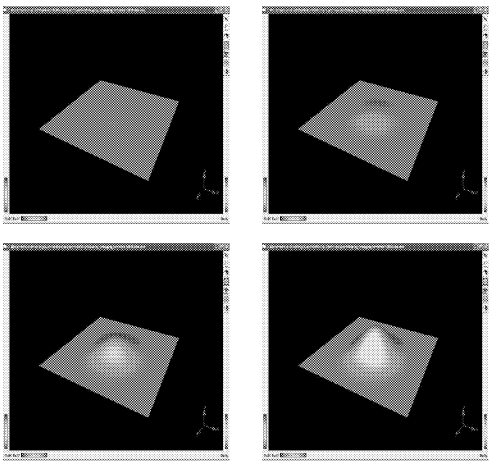


Figure 1: Dinamic simulation of a plane deformation using a positive constant force $(0, 100, 0)$.

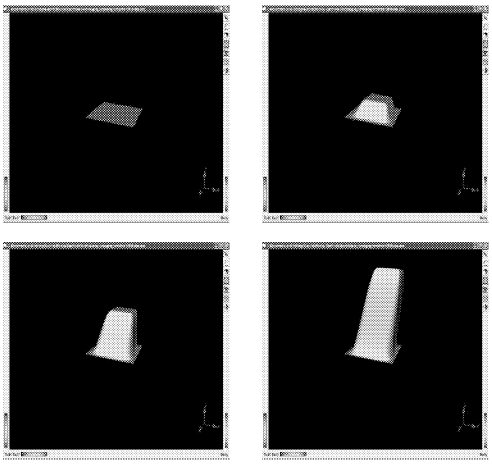


Figure 2: Dynamic simulation of a plane deformation using a force in the direction $(1, 4, 1)$

B-splines and $N_1 \times N_2 = 36$. We can compare this deformation with the obtained one in Figure 1.
More examples can be seen in González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007).

STUDY OF STABILITY

In order to study the stability of our numerical method (12), first of all, we need to write it as:

$$\begin{pmatrix} P_i(t+\Delta t) \\ P_i(t) \end{pmatrix} = \begin{pmatrix} M_1 & M_2 \\ \mathrm{Id} & 0 \end{pmatrix} \begin{pmatrix} P_i(t) \\ P_i(t-\Delta t) \end{pmatrix} + \begin{pmatrix} 2L_i\Delta t^2 \\ 0 \end{pmatrix}, \quad (14)$$

where M_1 and M_2 are $M_1 = (2M + \Delta tC)^{-1}(4M - 2A\Delta t^2)$, $M_2 = (2M + \Delta tC)^{-1}(\Delta tC - 2M)$.

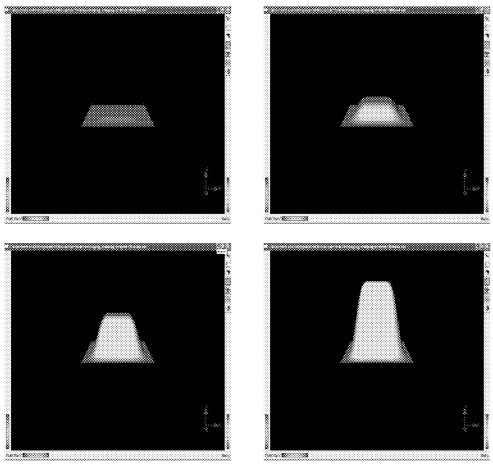


Figure 3: Several iterations of a plane deformation using the vertical force $(0, 200, 0)$ over all the surface.

Following Bathe (1982), the previous system will be stable if the spectral radius of the matrix

$$M = \begin{pmatrix} M_1 & M_2 \\ \mathrm{Id} & 0 \end{pmatrix},$$

is less than 1.
In order to study the stability, the expansion of eigenvalues λ of M as a function of Δt will be found. In this way, the behaviour of eigenvalues λ of M for Δt small will be studied.
Taking into account that $M = m$ Id and $C = c$ Id are diagonal matrices, M_2 is a diagonal matrix too. It can be written as: $M_2 = \frac{c\Delta t - 2m}{c\Delta t + 2m}\mathrm{Id}$. Let k be the constant $k = \frac{c\Delta t - 2m}{c\Delta t + 2m}$. So, $M_2 = k$ Id.
Let λ be an eigenvalue of M. It exists a vector x such that $Mx = \lambda x$. The last equation can be written as:

$$\begin{pmatrix} M_1 & k\mathrm{Id} \\ \mathrm{Id} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} M_1x_1 + kx_2 \\ x_1 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

where $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.
The previous system has a non-trivial solution if the determinant of the matrix $\lambda M_1 + (k - \lambda^2)$ Id is zero. That is, $\frac{k-\lambda^2}{\lambda}$ is an eigenvalue of M_1. (The case $\lambda = 0$ corresponds to take $\Delta t = \frac{2m}{c}$. So, for the experiments values $m = 1$ and $c = 0.3$, $\Delta t = 6.67$ is too large and this case cannot be taken into account.)
So, for every λ eigenvalue of M, there exists μ eigenvalue of M_1 such that

$$\lambda = \frac{\mu \pm \sqrt{\mu^2 + 4k}}{2}. \quad (15)$$

Taking into account that $M_1 = (2M + \Delta tC)^{-1}(4M - 2A\Delta t^2)$, for every eigenvalue μ of M_1, there exists an eigenvalue α of A such that $\mu = \frac{4m - 2\alpha\Delta t^2}{2m + c\Delta t}$.

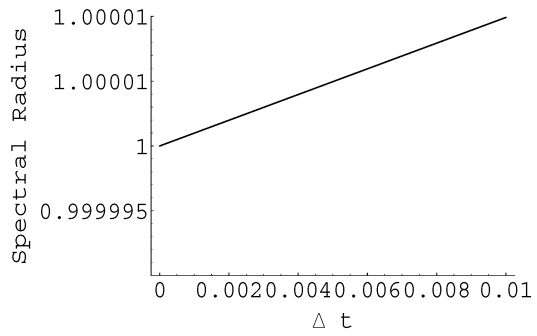


Figure 4: This graph shows the spectral radius of matrix M as a function of Δt

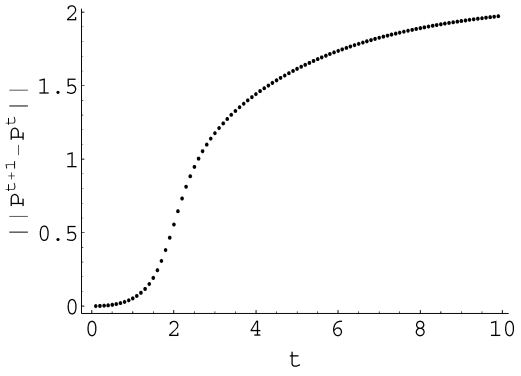


Figure 5: This graph shows the difference norm between two consecutive deformation control points from $t = 0$ to $t = 10$ seconds.

The expansion of eigenvalues μ of M_1 as a function of Δt is: $\mu = 2 - \frac{c}{m}\Delta t + \frac{1}{m^2}\left(\frac{c^2}{2} - \alpha m\right)\Delta t^2 + O\left(\Delta t^3\right)$. If we substitute this expression into equation (15), we obtain the expansion of eigenvalues λ of M:

$$\lambda = 1 - \left(c \mp \sqrt{c^2 - 4\alpha m}\right)\frac{\Delta t}{2m} + O\left(\Delta t^2\right).$$

So, if $\alpha < 0$ (α eigenvalue of A), our integration method is not stable. The computation of eigenvalues of A shows us that there exists negative eigenvalues of matrix A. Our system is not stable but the spectral radius is close to 1 as it can be seen if figure 4.

Our simulations show good deformations because the inestability of the numerical method is close to stability. In González-Hidalgo et al. (2006), it has been shown examples of good simulations using the numerical method. Moreover, the difference norm $\|P_{t+1} - P_t\|$ between two consecutive deformation control points isn't large as it can be seen in figure 5.

THE INFLUENCE OF FIRST ORDER PARAMETERS

In this section, we have studied how the simulations of the model can vary if the value of the parameters ω_{01} and ω_{10} are changed. To do this, we have made several deformations changing the value of the parameter ω_{01} first, and ω_{10} later. The other parameters and the applied force remain constant throughout the simulation. Let P^0 be the vector of control points of the initial surface; that is, the vector of control points of the surface before the deformation. The number of components of P^0 is $3 \times N_1 \times N_2$, three spatial components for each control point. So, if $P^0_1, \ldots P^0_{N_1 \times N_2}$ are the initial control points, P^0 can be written as:

$$P^0 = (P^0_{1,1}, P^0_{1,2}, P^0_{1,3}, \ldots, P^0_{N_1 \times N_2,1}, P^0_{N_1 \times N_2,2}, P^0_{N_1 \times N_2,3})^\top.$$

Using the same notation, let P^f be the vector of control points of the deformed surface. The influence of the parameters has been measured using the following norm:

$$||P^f - P^0|| = \sqrt{\sum_{i=1}^{N_1 \times N_2}\sum_{j=1}^{3}(P^f_{i,j} - P^0_{i,j})^2}. \qquad (16)$$

With the purpose of study the influence of the first order parameter, we have made several simulations with different values of them. The values of the second order parameters have been $\omega_{11} = 0.01, \omega_{20} = 0.01, \omega_{02} = 0.01$. The applied force in the simulations is zero over all surface except in a small central square whose value is 100. Figure 6 displays the linear approximation of the values, where horizontal axis is w_{10} or w_{01} and the vertical axis is the quadratic norm (16). The parameters values are 0; 0.0001; 0.001; 0.01; 0.1; 0.3; 0.5; 1; 2; 5; 7; 10; 25; 50; 75; 100; 150 and 200.

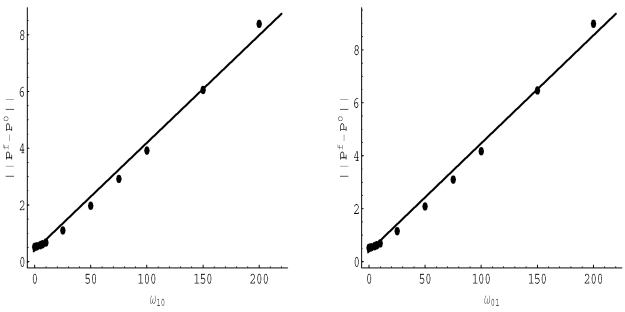
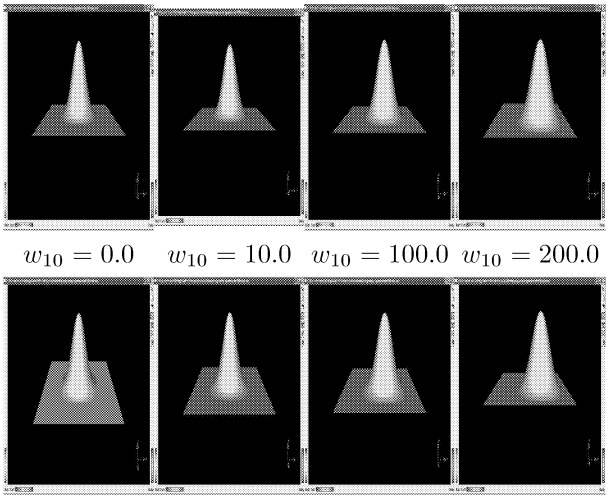


Figure 6: Left: graph of the norms (16) of the simulations vs ω_{10}. Right: graph of the norms (16) vs ω_{01}.

We can see in figure 6, as larger is the first order parameter, smaller is the resistance to the deformation. The relationship between the increase of the first order parameter and the decrease of the resistance to the deformation (16) is linear. The lines of regression are $\xi_1 = 0.4 + 0.038\,\omega_{10}$ and $\xi_2 = 0.3924 + 0.0407\,\omega_{01}$ where

ξ_1 and ξ_2 is the norm (16) with regression coefficient 0.992 for two parameters.
The effects made by this parameter can be seen in figure 7. This figure shows the effect of these parameters on a deformation made by a vertical force on a central square 0.2×0.2 and module 200. As the parameter increases, the length deformation increases too, following the direction of the applied force. So, we can conclude that the resistance to the length deformation decreases as the value of first order parameters increases.



$w_{10} = 0.0$ $w_{10} = 10.0$ $w_{10} = 100.0$ $w_{10} = 200.0$

$w_{01} = 0.0$ $w_{01} = 10.0$ $w_{01} = 100.0$ $w_{01} = 200.0$

Figure 7: last iteration of deformation with $\Delta t = 0.1$ and t=1. Vertical Force (0,200,0)

THE INFLUENCE OF SECOND ORDER PARAMETERS

To study the influence of second order parameters ω_{11}, ω_{02} and ω_{02}, a similar development to the first order parameters has been made. We have made several simulations with different values for these parameters. The values of the others parameters remain constant with value 0.1. We have applied a vertical force $(0, 100, 0)$ on a central square 0.2×0.2. In figure 8 the quadratic norms of the simulations with values $\omega_{ij} = 0$; $0.001; 0.01; 0.1; 1; 2; 5; 10; 25; 50; 75; 100; 125;$ $150;$ $175;$ 200 $(i + j = 2)$ are exposed. The regression line is also displayed. Figure 8 shows the following: as larger is the parameter ω_{ij} $(i + j = 2)$, smaller is the resistance to the deformation. The relationship between the increase of the resistance is linear.The regression lines are the following $\xi_3 = -0.00077475 + 0.00400753\,\omega_{11}$ for the parameter ω_{11}, $\xi_4 = 0.0041346\,\omega_{20} - 0.00122073$ for the parameter ω_{20} and $\xi_5 = 0.00410251\,\omega_{20} - 0.000905363$ for the parameter ω_{02} where ξ_3, ξ_4 and ξ_5 are the norm. The regression coefficients are larger than 0.99 for the three second order parameters.

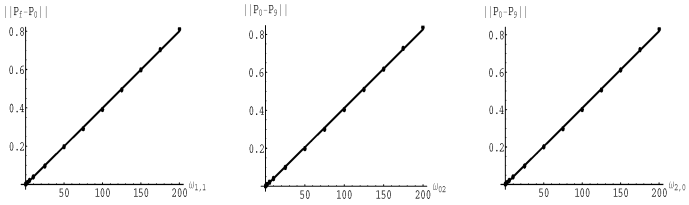
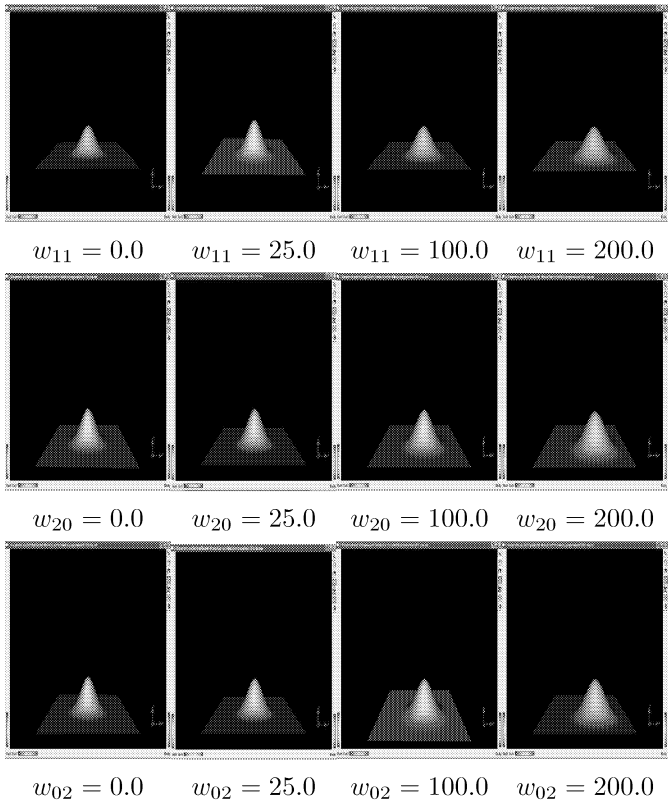


Figure 8: The quadratic norms and the regression line of the simulations vs the different second order parameters.

In figure 9 the last iteration for the parameter values 0, 25, 100 and 200 are displayed. The deformation is moved on plane XZ while the force is applied in direction Y.



$w_{11} = 0.0$ $w_{11} = 25.0$ $w_{11} = 100.0$ $w_{11} = 200.0$

$w_{20} = 0.0$ $w_{20} = 25.0$ $w_{20} = 100.0$ $w_{20} = 200.0$

$w_{02} = 0.0$ $w_{02} = 25.0$ $w_{02} = 100.0$ $w_{02} = 200.0$

Figure 9: last iteration of deformation with $\Delta t = 0.1$ and t=1. vertical force $(0, 100, 0)$

CONCLUSIONS AND FUTURE WORK

We have developed a variational formulation of a model that allows us the deformation of a surface. Moreover, we have shown how to solve the variational equations numerically in a efficient way using B-splines as finite elements. More explicitly, since only a certain number of control points is evolved, the numerical scheme obtained is very efficient.

Also, we have obtained a low computational cost, and we have shown that the model is nearly stable as it happens on the Terzopoulos model Palmer et al. (2000)
The study of behavior of first and second order parameters shows that the growth of these makes a linear decrease resistance of the surface, but in the case of second order parameters the decrease is slower than the first order parameters. In order to obtain a smooth deformation, the external force module must be small or the first order parameters must be smaller than 1. The implementation of this model has been made using C++ and Coin3D, a 3D modelling toolkit which simplifies visualization and scene composition tasks.
As future work, other numerical methods, as conjugate gradient or semi-implicit ones, will be applied. Moreover, other functionals will be taken into account in order to improve the deformation of the surface.

ACKNOWLEDGEMENTS

This work is supported by the project TIC-2004-07926-E, INEVAI3D, of the Spanish Government. The authors would like to thank to the department of Mathematics and Computer Science of University of the Balearic Islands.

REFERENCES

Bathe K.J., 1982. *Finite element procedures in engineering analysis*. Prentice-Hall Engineering and Engineering Mechanics Series. Prentice–Hall, Englewood Cliffs, New Jersey.

Cohen I., 1992. *Modèles Déformables 2-D et 3-D: Application à la Segmentation d'Images Médicales*. Ph.D. thesis, Université Paris IX, Dauphine.

Cox M.G., 1972. *The numerical evaluation of B-splines. IMA Journal of Applied Mathematic*, 10, no. 2, 134–149.

de Boor C., 1978. *A practical guide to splines*. Springer Verlag, New York.

Farin G., 1997. *Curves and Surfaces for Computer-Aided Geometric Desing: A Practical Guide*. Academic Press.

González M.; Mascaró M.; Mir A.; Palmer P.; and Perales F., 2001. *Modeling and animating deformable objects*. In *Proceedings of IX Spanish Symposium on Pattern Recognition and Image Analysis*. AERFAI Society, Benicasim, Castellón, Spain, 279–290.

González-Hidalgo M.; Mir A.; and Nicolau G., 2006. *An evolution model of parametric surface deformation using finite elements based on B-splines*. In *Proceedings of CompImage'2006 Conference, Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*. Coimbra, Portugal.

González-Hidalgo M.; Mir A.; and Nicolau G., 2007. *Dynamic parametric surface deformation using finite elements based on B-splines. To appear in International Journal of Computer vision and Biometricss*, 1, no. 1.

Höllig K., 2003. *Finite element methods with B-Splines*. Frontiers in Applied Mathematics. SIAM, Philadelphya.

Mascaró M., 2002. *Modelo de Simulación de Deformaciones de Objetos Basado en la Teoría de la Elasticidad*. Ph.D. thesis, Universitat de les Illes Balears.

Mascaro M.; Mir A.; and Perales F., 2002. *P^3DMA : A physical 3D deformable modelling and animations system. LNCS*, 2492, 68–79.

Montagnat J.; Delingette H.; and Ayache N., 2001. *A review of deformable surfaces: topology, geometry and deformation. Image and Vision Computing*, 19, no. 14, 1023–1040. URL `citeseer.nj.nec.com/498203.html`.

Palmer P.; González M.; and Mir, 2000. *Stability and complexity study of animated ellastically deformable objects*. In *Proceedings of I Articulated Motion and Deformable Objects*. Palma de Mallorca Spain, 58–71.

Piegl L. & Tiller W., 1997. *The NURBS book*. Springer Verlag, Berlin.

Qin H & Terzopoulos D., 1997. *Triangular nurbs and their dynamic generalizations. Computer Aided Geometric Design*, 14, 325–347.

Raviart P. A. & Thomas J.M., 1992. *Introduction à l'analyse numérique des équations aux derivées partielles*. Masson, Paris.

Terzopoulos D., 1986. *Regularization of inverse visual problems involving discontinuities. IEEE PAMI*, 8, no. 4, 413–424.

STRUCTURAL ANALYSIS OF DAE MODELS

Adrien Leitold
Department of Mathematics
University of Pannonia
P.O.B. 158, Veszprém, H-8201,
Hungary
E-mail: leitolda@almos.vein.hu

Miklós Gerzson
Department of Automation
University of Pannonia
P.O.B. 158, Veszprém, H-8201,
Hungary
E-mail: gerzson@almos.vein.hu

KEYWORDS
process models, DAE-models, differential index, solvability, structural analysis

ABSTRACT

A graph-theoretical method for the structural analysis of dynamic lumped process models described by differential and algebraic equations (DAEs) is applied in this paper in order to determine the most important solvability properties of these models by using the so-called dynamic representation graph. In our earlier works we have introduced the graph representation of first order, semi-explicit DAEs and the investigation method of the solvability properties (degree of freedom, structural solvability, model decomposition, dynamic degree of freedom, differential index, e.g.) based on the structure of the resulting dynamic representation graph. In our recent work we investigate graph representation of DAE systems containing higher order and/or implicit differential equations. The structure of dynamic representation graph is suitable for the determination of the mentioned solvability properties in case of this type of systems and the determination of the differential index is shown as an example.

INTRODUCTION

The structural analysis of dynamic lumped process models forms an important step in the model building procedure (Hangos and Cameron 2001) and it is used for the determination of the solvability properties of the model, too. This analysis includes the determination of the degree of freedom, structural solvability, differential index and the dynamic degrees of freedom. As a result of the analysis, the decomposition of the model is obtained and the calculation path can be determined. This way the appropriate numerical method for solving the model can be chosen efficiently. Moreover, advice on how to improve the computational properties of the model by modifying its form or its specification can also be given.

Effective graph-theoretical methods have been proposed in the literature (Leitold and Hangos 2001 and 2002) based on the analysis tools developed by (Murota 1987), for the determination of the most important solvability property of lumped dynamic models: the differential index. The properties of the dynamic representation graph of process models described by semi-explicit DAE-systems have also been analysed there in case of index 1 and higher index models. Beside the algorithm of determining the differential index by using the representation graph, a model modification method has also been proposed in the literature, which results in a structurally solvable model even in the case of higher index models (Leitold and Hangos 2001).

BASIC NOTIONS

Structural Solvability

As a first step, we consider a system of linear or non-linear algebraic equations in its so called *standard form* (Murota, 1987):

$$y_i = f_i(x, u), \qquad i = 1, \ldots, M$$
$$u_k = g_k(x, u), \qquad k = 1, \ldots, K$$

where x_j ($j = 1,\ldots, N$) and u_k ($k = 1,\ldots,K$) are unknowns, y_i ($i = 1,\ldots, M$) are known parameters, f_i ($i = 1,\ldots, M$) and g_k ($k = 1,\ldots, K$) are assumed to be sufficiently smooth real-valued functions. The system of equations above is structurally solvable, if the Jacobian matrix $J(x, u)$ referring to the above model is non-singular.

Consider a system of equations in standard form. We construct a directed graph to represent the structure of the set of equations in the following way. The vertex-set corresponding to unknowns and parameters is partitioned as $X \cup U \cup Y$, where $X = \{x_1, \ldots, x_N\}$, $U = \{u_1, \ldots, u_K\}$ and $Y = \{y_1, \ldots, y_M\}$. The functional dependence described by an equation is expressed by arcs coming into y_i or u_k respectively from those x_j and u_l, which appear on its right-hand side. This graph is called the *representation graph* of the system of equations.

A *Menger-type linking* from X to Y is a set of pair-wise vertex-disjoint directed paths from a vertex in X to a vertex in Y. The size of a linking is the number of directed paths from X to Y contained in the linking. In case $|X| = |Y|$, ($M = N$), a linking of size $|X|$ is called a *complete linking*. The graphical condition of the structural solvability is then the following (Murota 1987):

Linkage theorem: Assume that the non-vanishing elements of partial derivatives f_i and g_k in the standard form model are algebraically independent over the rational number field Q. *Then the model is structurally solvable if and only if there exists a Menger-type complete linking from X to Y on the representation graph.*

We can adapt the graphical techniques to DAE-systems, as well. An ordinary differential equation of a DAE-system can be described by the following equation:

$$x' = f(x_1, \ldots, x_n)$$

Here x denotes an arbitrary variable depending on time, x' denotes the derivative of x with respect to time and x_1, …, x_n are those variables which have effect on variable x' according to the differential equation.

In DAE-systems there are two types of variables. *Differential variables* are the variables with their time derivative present in the model. Variables, which do not have their time derivative present, are called *algebraic variables*. The derivative x' is called *derivative* (velocity) *variable*.

Dynamic Representation Graph

The value of differential variables is usually computed by using a numerical integration method. Therefore a system of equations including also differential equations can be represented by a *dynamic graph*. A dynamic graph is a sequence of static graphs corresponding to each time step of the integration. On a dynamic graph there are directed arcs attached from the previous static graph to the succeeding static graph that are determined by the method applied for solving the ordinary differential equations. In case of a single step explicit method, the value of a differential variable at time $t+h$ is computed using the corresponding differential value and its value at a previous time t. For example, when the explicit Euler method is used:

$$x(t+h) = x(t) + h \cdot x'(t)$$

where h denotes the step length during the numerical integration. The structure of a dynamic graph assuming explicit Euler method for solving differential equations is shown in Figure 1.

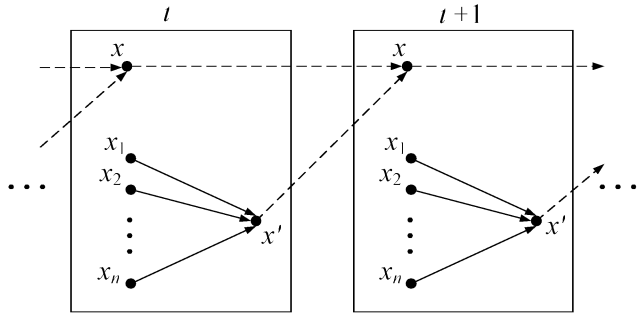


Figure 1: Dynamic Representation Graph Assuming First Order Explicit Solution Method

The structural analysis based on graph theoretical technique is carried out in steps performed sequentially. The first step is to rewrite the model into its standard form. The second step is the assignment of types to vertices in the representation graph. The important types of vertices determined by the model specification are the following (Iri et al. 1972; Leitold and Hangos 2001):

- <*S*>(*set*)*-type variables*: These represent variables, which are assigned to the specified given values. In the case of a dynamic representation graph assuming explicit method for solving the differential equations, the differential variables will be labelled by type <*S**> because their initial value can be obtained from the initial values, and then their values can be calculated step by step by numerical integration. Labels <*S*> and <*S**> are treated the same way during the analysis.
- <*G*>(*given*)*-type variables*: A variable assigned to a specific value of a left hand side is a <*G*>-type variable. Unlike the <*S*>-type variables, the values of the right hand side variables will be suitably adjusted so as to preserve the equality of the two sides.

According to the representation graph, the value of every variable which has incoming arcs only from vertices labelled by type <*S*> can be calculated by simple substitution into the corresponding equation. These variables become *secondarily labelled by type* <*S*>, and this process can be repeated if necessary. Omitting all vertices labelled primarily, secondarily, etc. by type <*S*> and all arcs starting from them from the representation graph we obtain the *reduced graph*. The classification of vertices of a reduced graph is as follows:

- all initial vertices form the unknown variable set X,
- all terminal vertices labelled by type <*G*> constitute the known variable (parameter) set Y,
- all other vertices constitute the unknown variable set U.

Differential Index

Dynamic process models can be described by *semi-explicit DAEs* as follows:

$$z_1' = f(z_1, z_2, t), \quad z_1(t_0) = z_{10} \qquad (1)$$

$$0 = g(z_1, z_2, t) \qquad (2)$$

The most important structural computational property of DAE models is the differential index. By definition (Brenan, et al. 1989) the differential index of the semi-explicit DAE (Equations (1)-(2)) is one if one differentiation is sufficient to express z_2' as a continuous function of z_1, z_2 and t. One differentiation is sufficient if and only if the Jacobian matrix g_{z_2} is non-singular.

In our earlier work we have proved that the differential index of the models investigated in (Leitold and Hangos 2001) *is equal to 1 if and only if there exists a Menger-type complete linking on the reduced graph at any time step t.*

If the differential index of the investigated model is greater than 1 then there is no Menger-type complete linking on the static graph at any time step t. The properties of a static graph of a dynamic model, which has differential index >1 are as follows.

1. The fact that the initial values of differential variables cannot be chosen independently results in an *overspecified* part on the graph. This situation can be easy shown by assignment of types to vertices corresponding to the model specification. There is an overspecified part on the graph if a vertex labelled by type <*S**> or <*G*> can also be labelled secondarily or tertiarily or etc. by type <*S*>.

2. Non-singularity of g_{z_2} results in an *underspecified* part on the graph. In this part those algebraic variables appear, which cannot be calculated from algebraic equations and those differential variables, which we want to calculate from them.

We have also proposed an algorithm using the structure of the representation graph for determination of the differential index of the underlying model. The main steps of this algorithm are the following:

1. Let us form the following variable sets.
   I_0 is the set of the differential variables belonging to the overspecified subgraph,
   D_0 is the set of the derivative variables referring to the differential variables of set I_0,
   I_1 is the set of differential variables from which directed paths lead to the derivative variables in the set D_0,
   D_1 is the set of derivative variables referring to the differential variables of set I_1, … ,
   I_k is the set of differential variables from which directed paths lead to the derivative variables in the set D_{k-1},
   D_k is the set of derivative variables referring to the differential variables of set I_k, …
2. Let n be the smallest natural number for which the set D_n contains some derivative variables of the underspecified subgraph. Then the differential index of the model is
   $$\nu_d = n+2$$
   If there is no such number n then the model is not structurally solvable.

In our earlier work we have shown that the important properties of the representation graph including the differential index of the models are independent of the assumption whether a single-step, explicit or implicit numerical method is used for the solution of the differential equations (Leitold and Hangos 2005).

REPRESENTATION OF DAE SYSTEMS CONTAINING HIGHER ORDER DIFFERENTIAL EQUATIONS

Models of mechanical systems often contain higher order differential equations. In order to transform these models to DAE systems containing first order differential equations we introduce new variables. The method described in the previous section can be used for deriving the representations of the transformed model and performing its structural analysis. We illustrate the graph representation and the determination of the differential index using the following two examples.

Example 1. – Pendulum

Pendulum is one of the most well known example of high index model of mechanical system (Brenan et al. 1989).

The pendulum consists of a body connected to a bar. Let the mass of the body (m) be equal to 1 and the length of the bar (L) be equal to 1, too. λ denotes the drag force in the bar, and g the gravity force, respectively. The mathematical model is the following:

$$x'' = \lambda x \qquad x'(t_0) = x'_0 \qquad x(t_0) = x_0$$
$$y'' = \lambda y - g \qquad y'(t_0) = y'_0 \qquad y(t_0) = y_0$$
$$0 = x^2 + y^2 - 1$$

The model consists of two second order differential equations (Newton's second law) and one algebraic equation (description of the path of the body). The dynamic degree of freedom of the model is 2, and the model is semi-explicit index 3.

A model containing higher order differential equations can be transformed into a DAE system containing only first order differential equations. Let us introduce the following new variables in the pendulum example:

$$v_x = x' \qquad v_y = y'$$

The standard form of pendulum example is the following:

$$x = \int x' \qquad y = \int y'$$
$$v_x = \int v_x' \qquad v_y = \int v_y'$$
$$x' = v_x \qquad v_x' = \lambda x$$
$$y' = v_y \qquad v_y' = \lambda y - g$$
$$s = x^2 + y^2 - 1, \quad s = 0$$

Given: $x(t_0)$, $y(t_0)$, $v_x(t_0)$, $v_y(t_0)$;

Constant: g;

To be calculated: x, y, v_x, v_y, λ as functions of time.

The dynamic representation graph of the pendulum model is shown in Figure 2.

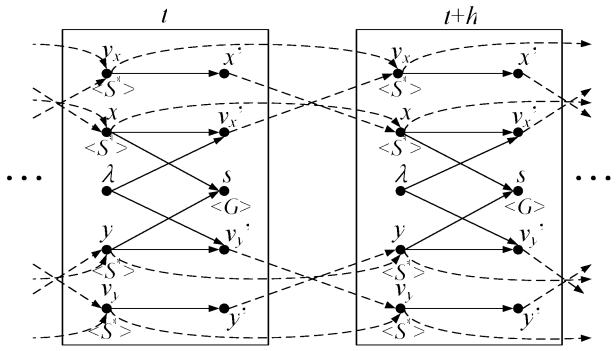


Figure 2: Representation Graph of Pendulum Model

An overspecified and an underspecified subgraph (see Figure 3) can be found on the static graphs referring to the greater than 1 value of the differential index. The differential index can be calculated based on the structures of these overspecified and underspecified subgraphs:

$$I_0 = \{x, y\}$$
$$D_0 = \{x', y'\}$$
$$I_1 = \{v_x, v_y\}$$
$$D_1 = \{v_x', v_y'\}$$

Since the vertices referring to differential variables v_x' and v_y' can be found in the underspecified subgraph, $n = 1$ and $\nu_d = n + 2 = 3$.

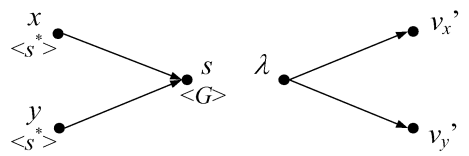


Figure 3: The Overspecified and Underspecified Subgraphs on the Representation Graph of Pendulum Model

Example 2. – PID Controller

As a second example let us take the following PID controller. A horizontal force F is applied to a body of mass m on flat surface, without friction. The position of the body is denoted by x. The control objective is to keep the body at a given position x_{set}. The mathematical model of the control system is the following:

$$x'' = (F - u)/m$$
$$i' = e$$
$$e = x - x_{set}$$
$$u = k_P e + k_D e' + k_I i$$

The unknowns are x, v, i, e, u; variable F is an input variable (depending only on time), and x_{set}, m, k_P, k_D, k_I are constants. The resulting model is an index 2 system of DAEs.

Introducing the variable $v = x'$ the model can be transformed into the following standard form:

$$x = \int x' \qquad v = \int v'$$
$$i = \int i' \qquad e = \int e'$$
$$x' = v \qquad v' = (F - u)/m$$
$$i' = e \qquad e = x - x_{set}$$
$$e' = (u - k_P e - k_I i)/k_D$$

Given: $x(t_0)$, $v(t_0)$, $i(t_0)$, $e(t_0)$, $F(t)$, m, x_{set}, k_P, k_D, k_I;
To be calculated: x, v, i, e, u as functions of time.

The representation graph belonging to the model of PID control system is shown on Figure 4.

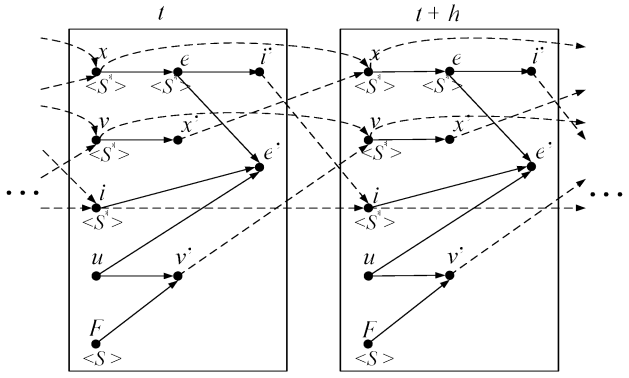


Figure 4: Representation Graph of PID Controller Model

There are overspecified and underspecified subgraphs on the static graphs (see Figure 5) and based on their structures the differential index can be calculated:

$$I_0 = \{x, e\}$$
$$D_0 = \{x', e'\}$$

Since the vertex referring to differential variable e' appears in the underspecified subgraph, $n = 0$ and $\nu_d = n + 2 = 2$.

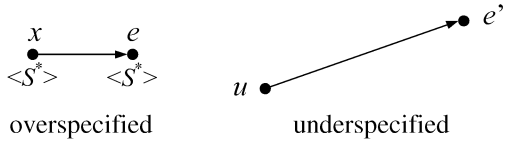


Figure 5: The Overspecified and Underspecified Subgraphs of PID Controller Model

REPRESENTATION OF DAE SYSTEMS CONTAINING IMPLICIT FORM DIFFERENTIAL EQUATIONS

Rearranging the output equation of PID controller into following implicit form:

$$u = k_P e + k_D e' + k_I i \ ,$$

the resulting equation is an implicit form differential equation.

Applying the nonlinear form of the implicit Euler method with time step h for solving the implicit differential equation $F(t, x, x') = 0$, the following form is obtained:

$$F\left(t_n, x_n, \frac{x_n - x_{n-1}}{h}\right) = F\left(t_n, x_n, x_n'\right) = 0 \qquad (3)$$

This equation can be solved with a nonlinear equation solver method.

Based on this method an implicit differential equation can be represented as it is shown in Figure 6. In this figure x_1, …, x_k refer to other variables occurring on the left side of the implicit differential equation. The arrows with solid lines represent the structure of differential equation, while the arrows with dashed lines are coming from the numerical solution of implicit differential equation. The auxiliary variable s_x refers to the 0 in the right side of Equation (3).

Starting form the initial value of differential variable x and from the equation $s_x = 0$, the values of variables x and x' can be determined step by step based on the Equation (3) during the numerical solution and this is the reason why vertices x and x' of the static graphs are denoted by label <S*> in the representation.

If there are underspecified and overspecified subgraphs on the static graphs the way of the determination of differential index has to be modified. After generating the sets of variables I_k and D_k ($k = 0, 1, \ldots$) *let n be the smallest natural number for which the set D_n contains one of the derivative variable x' of the underspecified subgraph or its auxiliary variable s_x. Then the differential index of the model is* $\nu_d = n+2$.

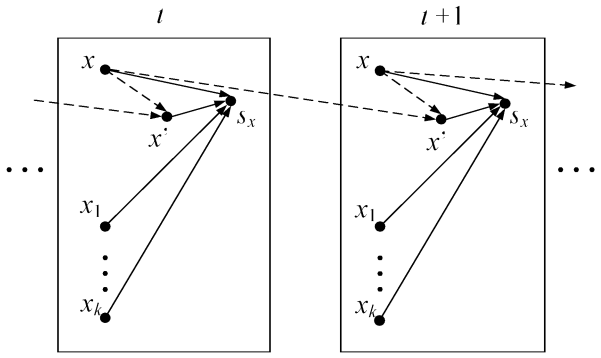


Figure 6: Dynamic Representation Graph of an Implicit Differential Equation

Example 3.

As an illustration of the described method let us consider the implicit form of the differential equation relating to variable *e'* in the model of PID controller:

$$s_e = k_P e + k_D e' + k_I i - u, \quad s_e = 0$$

The representation graph of the modified model is shown in Figure 7. Similarly to the Figure 4 overspecified and underspecified subgraphs appear on the representation graph (see Figure 8). From their structure the differential index v_d is equal to 2 based on the modified determination method.

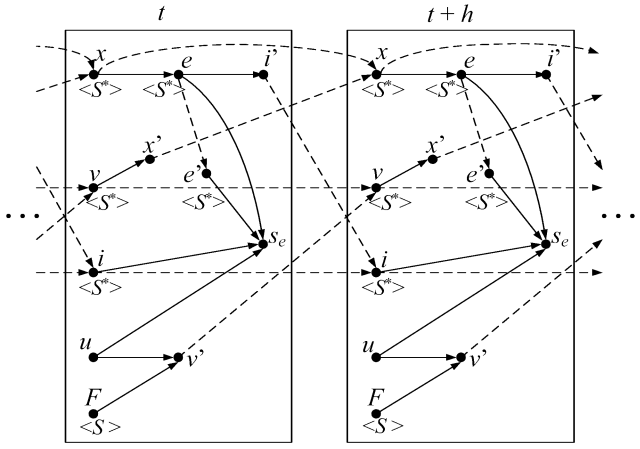


Figure 7: Representation Graph of PID Controller Model Containing Implicit Form Differential Equations

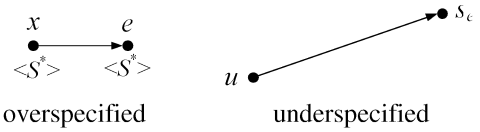


Figure 8: The Overspecified and Underspecified Parts of Representation Graph Shown in Figure 7

CONCLUSION

In this paper we investigated the structural analysis of dynamic models described by DAE systems using dynamic representation graphs. We applied our earlier developed method to DAE systems containing higher order and/or implicit form differential equations. We have shown by deriving the differential index, that the structure of the representation graph is suitable for the determination of the structural properties.

REFERENCES

Brenan, K.E., Campbell, S.L. and Petzold, L.R. 1989. *Numerical Solution of Initial Value Problems in Differential – Algebraic Equations*. North-Holland, New York

Hangos, K.M. and Cameron, I.T. 2001 *Process Modelling and Model Analysis*. Academic Press, London

Iri, M., Tsunekawa, J. and Yajima, K. 1972. "The Graphical Techniques Used for a Chemical Process Simulator 'JUSE GIFS'", *Information Processing*, 71. Vol. 2. 1150-1155

Leitold, A. and Hangos, K.M. 2001. "Structural Solvability Analysis of Dynamic Process Models." *Computers chem. Engng* 25, 1633-1646

Leitold, A. and Hangos, K.M. 2002. "Effect of Steady State Assumption on the Structural Solvability of Dynamic Process Models." *Hung. J. of Ind. Chem.* 30 1 61-71

Leitold, A. and Hangos, K.M. 2005. "Numerical Method-independent Structural Solvability Analysis of DAE Models." *Hung. J. of Ind. Chem.* 33 1-2 11-21

Murota, K. 1987. *Systems Analysis by Graphs and Matroids*. Springer Verlag, Berlin

AUTHOR BIOGRAPHY

ADRIEN LEITOLD received the M.Sc. degree in chemical engineering from the University of Pannonia, Veszprém and the M.Sc. degree in mathematics education from the University of Debrecen. She received her PhD degree in informatics from the Eötvös Lóránd University, Budapest. She is currently associate professor in the Department of Mathematics and Computing at the University of Pannonia, Veszprém, Hungary, where she gives lectures on linear algebra, operation research, theory of algorithms and complexity.

MIKLÓS GERZSON received the M.Sc. degree in chemical engineering from the University of Pannonia, Veszprém. He received his PhD degree in chemical engineering from the Hungarian Academy of Science. He is currently associate professor in the Department of Automation at the University of Pannonia, Veszprém, Hungary, where she gives lectures on control theory, measurement theory and technique.

MODELLING APPLICATIONS

Reducing the Power Factor Penalty of the Industrial Sector Due to Jordanian Tariff

NABEEL TAWALBEH
Isra Private University
Electrical Engineering Department
Amman - Jordan
nabiltawalbeh@yahoo.com

AYMAN AGHA
Jordan Phosphate Mines Co
Projects' Department
Amman - Jordan
aay@o2.pl

INTRODUCTION

In industrial plants, where usually operate under various electrical loads with various power factor (PF), the normal encountered PF varies in general [1] between 0, 50 and 0, 75.

In Jordan, the Electricity Regulatory Commission (ERC), basing on the tariffs structure impose a penalties on the consumers of the electrical energy whose the average value of the PF becomes lower than a cretin value, the additional payments to the electricity bills as penalties are stated in special tariffs [2]. To maintain the average value of the PF within a certain category limit and to reduce the penalties, the consumption of the active energy (E_p) shall be controlled. This can be done by consuming an extra amount of active Energy (E_{EX}), without changing pattern of the reactive power consumption

PROBLEM FORMULATION

To estimate the Total Bill Value (TBV), the PF and the power factor penalty category ($PF_{pen\text{-}ctg}$). A few days before the end of the meters' reading period of the energy, which is usually one month, the readings of active energy E_{p1} and reactive energy E_{q1}meters shall be taken out. Based on the data obtained, the TBV may estimated for the PF and ($PF_{pen\text{-}ctg}$) till the end of the meters' reading period. If the PF and the ($PF_{pen\text{-}ctg}$) are within the acceptable values and category, then the TBV and the PF shall be controlled to the end of the cashing period. However if an adjustment required to bring the ($PF_{pen\text{-}ctg}$) to the lower one, then an extra amount of active power P_{EX} must be consumed, without the changing pattern of the consumption of the reactive power (Q). The illustration of the above is demonstrated in Fig.1 and explained in the following mathematical equations.

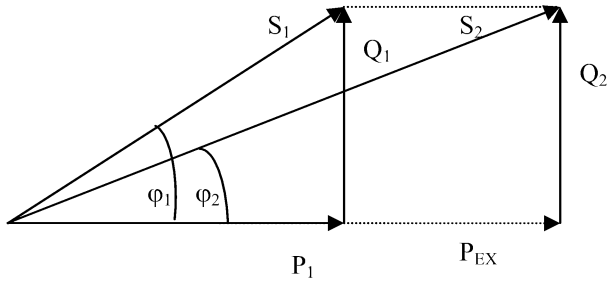


Figure 1. Illustrating the shifting in the power factor

$$PF_1 = Cos\varphi_1 = \frac{P_1}{S_1} = \frac{P_1}{(P_1^2 + Q_1^2)^{1/2}} \quad (1)$$

Considering,

$$P_2 = P_1 + P_{EX} \quad (2)$$

Obtaining

$$PF_2 = Cos\varphi_2 = \frac{P_2}{S_2} = \frac{P_2}{\left[(P_2)^2 + (Q_2)^2\right]^{1/2}} \quad (3)$$

Whereas, the Q pattern before and after the achieving of the required ($PF_{pen\text{-}ctg}$) remain constant, (i.e. $Q_1 = Q_2$).

From Fig 1 and eq (2) may obtain eq (4).

$$P_1 \tan\varphi_1 = P_2 \tan\varphi_2 \quad (4)$$

The integration of eq (4) over the period of consumed energy (the working time, T_w), the consumed energy (E_p) may given by:

$$E_{P1} \tan\varphi_1 = E_{P2} \tan\varphi_2 \quad (5)$$

The amount of consumed energy E_{p1} is taken out from the energy meters readings a few days before the meters reading period. To calculate the amount of active energy required to shift ($PF_{pen\text{-}ctg}$) to the lower category, eq (4) with respect to E_{p2} shall be solved.

$$E_{P2} = \frac{E_{P1} \tan\varphi_1}{\tan\varphi_2} \quad (6)$$

Take in mind $\tan\varphi = \frac{\sin\varphi}{\cos\varphi}$; *and*, $\sin\varphi = \sqrt{1 - \cos^2\varphi}$

yields

$$E_{P2} = E_{P1} \times \frac{\cos\varphi_2}{\cos\varphi_1} \frac{\sqrt{1 - \cos^2\varphi_1}}{\sqrt{1 - \cos^2\varphi_2}} \quad (7)$$

Where; E_{p2} is the amount of active energy required to bring the PF level from the higher level of power factor penalty category ($PF_{pen\text{-}ctg}$) to the lower one [3]. An extra amount of active energy E_{EX} that, shall be consumed to achieve the required adjustment is as in eq (8)

$$E_{EX} = E_{P2} - E_{P1} \quad (8)$$

And in percentage (%) w.r.t Ep1 as in eq (9)

$$E_{EX} = \frac{E_{P2} - E_{P1}}{E_{P1}} \times 100\% \quad (9)$$

The estimated amount in (JD) of the total bill value $TBV_{(est)}$ over the accounted period can be expressed as follows:

$$TBV_{(est)} = C_m \times MD + C_{pd} \times E_{pd(est)} + C_{pn} \times E_{pn(est)} + PF_{pen(est)} + C_r \times E_{p(est)} + C_{nr} \quad (10)$$

And, the estimated amount of the of power factor penalties in (JD) could also be expressed as follows: ($PF_{pen\text{-}ctg}$)

$$PF_{pen(est)} = (PF_{pen-ctg}) \times (C_{pd} \times E_{pd(est)} + C_{pn} \times E_{pn(est)} + C_m \times MD) \times \Delta\cos \quad (11)$$

Where:

C_m Maximum load tariff is JD/kW/Month

C_{nr} Fees not related to consumed energy JD/Month

C_{pd} Day energy tariff JD/kWh

C_{pn} Night energy tariff JD/kWh

C_r Fees and Taxes related to consumed energy JD/kWh

$PF_{pen.}$ Power factor penalty value JD/Month

$\Delta\cos\varphi$ The violation of the power factor below the penalty limit, ($\Delta\cos\varphi = 0.85 - \cos\varphi_{avg.}$); and COS $\varphi_{avg.}$ – average value; COS $\varphi_{avg\,(rqd)}$ – required value of COS φ.

The $PF_{pen\text{-}ctg}$ according to [1] is as follows;

- For PF equal 0.85 or more - No Penalty.
- For PF Less than 0.85 to 0.70, the $PF_{pen\text{-}ctg}$ is 0.77 % of For total bill for every 0.01 of the PF less than 0.85
- The PF Less than 0.70 To 0.60, the $PF_{pen\text{-}ctg}$ is 0.95 % of the total bill for every 0.01 of the PF less than 0.85
- For PF Less than 0.60 To 0.50, the $PF_{pen\text{-}ctg}$ is 1.2 % of the total bill for every 0.01 of the PF less than 0.85.
- For PF Less than 0.50, the $PF_{pen\text{-}ctg}$ is 1.5 % of the total bill for every 0.01 of the PF less than 0.85.

Generally, the cost of 1 kWh during night tariff is less than the day tariff. Therefore, to achieve maximum benefits from eq (10), it is recommended to use the E_{EX} during the night tariff as possible and to exclude the peak load (MD) period if encountered.

The cost of the extra energy E_{EX} in (JD) is equal to:

$$E_{EX(cost)} = C_{pd} \times E_{pd(EX)} + C_{pn} \times E_{pn(EX)} \quad (12)$$

Where:

$$E_{pd(EX)} = P_{EXd} \times T_{EXd} \quad (13)$$

$$E_{pn(EX)} = P_{EXn} \times T_{EXn} \quad (14)$$

Where:

P_{EXd} - Extra amount of active power, consumed during the time T_{EXd} at the day tariff.

P_{EXn} - As above but for the night tariff.

T_{EXd} - Remaining (hours) period to the end of the measured period at Day tariff.

T_{EXn} - Remaining (hours) period to the end of the measured period at Night tariff.

t_{EXd} - Daily working hours time at Day tariff excluding the peak load period.

t_{EXn} - Daily working hours time at Night tariff excluding the peak load period.

N_d - Number of the working days remaining to the end of the measured period.

Utilizing the eq (13) & (14), we have to consider that, the apparent power S_2 shall be less than the system equipment's capability and P_{EX} shall be less equal than the free amount of active power available in the system.

The estimated amount in (JD) of the TBV over the accounted period after consuming the E_{EX} can be expressed as follows:

$$TBV_{(after)} = C_m \times MD + C_{pd} \times E_{pd(est)} + C_{pn} \times E_{pn(est)} + E_{EX(\cos t)} + PF_{pen(est)} + C_T \times (E_{p(est)} + E_{EX}) + C_{nr} \quad (15)$$

Where:

$$PF_{Pen(est)} = PF_{pen-ctg.} \times [C_{pd}.(E_{pd(est)} + E_{pd(EX)}) + C_{pn} \times (E_{pn(est)} + E_{pn(EX)}) + C_m.MD] \times \Delta\cos\varphi_{avg(rqd)} \quad (16)$$

The TBV, for the meters reading periods before and after the adjustment shall be compared and if the inequality eq (17) is realized then a saving is achieved;

$$TBV_{(after)} > TBV_{(est)} \quad (17)$$

The eq (17) shall be controlled until the end of the measured period. However, if the result eq (17) is not satisfactory, then there is no need to adjust the balance by active energy.

Numerical Results

To examine the implementation of the above-mentioned approach, electrical bills of an industrial plant were tested. The plant is working on three shifts. The electrical energy consumption and the PF level over the months are waving. In general, the encountered average value of the PF is less than the allowable one by the EA. A penalty due to that is imposed. The effects of saving were investigated. A check reading of the energy meters before the end the measured period is taken and checked and evaluated. The results are presented in table 1.

Table 1. The total bill value – before and after the adjustment of the Power Factor

Period	TBV [JD]	E_p [kWh]	PF_{pen} [JD]	Cosφ [Avg]	E_{EX} [kwh]	E_{EX} [JD]	Sav. [JD]
			Case I				
Check Reading	91,911	1,512,633	12,554	0.673			
$TBV_{(est)}$	111,703	1,973,000	15,204	0.673			
$TBV_{(after)}$	111,756	2,089,863	10,905	0.7	116,863	4,351	- 53 *
			Case II				
Check	78,350	1,287,5	10,23	0.682			

Reading		40	0				
Period	TBV [JD]	E_p [kWh]	PF_{pen} [JD]	Cosφ [Avg]	E_{EX} [kwh]	E_{EX} [JD]	Sav. [JD]
$TBV_{(est)}$	99,974	1,684,000	12,975	0.683			
$TBV_{(after)}$	93,054	1,694,844	9,092	0.7	10,844	404	935
		Case III					
Check Reading	83,017	1,291,067	10,815	0.683			
$TBV_{(est)}$	99,974	1,684,000	12,975	0.683			
$TBV_{(after)}$	99,040	1,746,291	9,696	0.7	62,291	2,346	934
		Case IV					
Check Reading	91,025	1,549,433	11,499	0.688			
$TBV_{(est)}$	111,224	2,021,000	14,004	0.688			
$TBV_{(after)}$	109,912	2,073,561	10,720	0.7	52,561	1,972	1,313
		Case V					
Check Reading	115,204	1,614,600	14,382	0.692			
$TBV_{(est)}$	135,996	2,106,000	16,903	0.692			
$TBV_{(after)}$	133,597	2,142,401	13,151	0.7	36,401	1,353	2,399
		Case VI					
Check Reading	77,437	1,287,540	9,316	0.697			
$TBV_{(est)}$	93,990	1,679,400	11,268	0.697			
$TBV_{(after)}$	92,123	1,690,244	8,997	0.7	10,844	404	1,867

For the above values, it is assumed that, the check readings are taken in a period of one week before the end of the meters reading period.

*** (-)** Sign minus means no saving in money.

The analysis of the above points out, there is a benefit in money saving when the adjustment of the consumed active energy is around the upper limit of the penalty category and vies versa.

RESULTS & RECOMMENDATIONS

This paper presents the influence of the power factor penalty category ($PF_{pen\text{-}ctg}$) on the Total electricity Bill Value (TBV). By consuming an extra amount of active energy, the ($PF_{pen\text{-}ctg}$) can change and this can lead savings in money in the TBV or allow to consume more energy by paying less (as it appeared in case 1 from the Numerical Results).

The extra amount of the active energy (E_{EX}) is normally determined in an enough time prior to the date of meters reading electricity meters' reading. It is assumed that, a period of one week prior to the meters reading is sufficient to check the ability to control the power factor category and to keep it under control.

Finally, it is understood that, nothing can substitute the use of the capacitor banks in compensating the reactive power, to correct the power factor and to achieve the benefits arising from the compensation, however the purpose of discussion is only to highlight the erroneous in the existing PF penalty tariff structure and to encourage the utilities to use another formula [4, 5 & 6] for the power factor penalty. Moreover, the tariff policy must be changed, to increase the penalty of decreasing the power factor. Otherwise some utility on the industrial sector may escape from one PF category to another with more consuming energy, with out taken into account the limitation of the source of energy.

REFERENCES

[1] Agha Ayman, "Probabilistic approach for reactive power compensation", Dirasat, Engineering Sciences Magazine, Vol. 26, No.2, Oct. 1999 Amman- Jordan.

[2] Electricity Tariff Instructions in Jordan" Effected from 9th, 2005. "official Magazine

[3] Agha Ayman, Nabeel Tawalbeh **"**Economic Discussion on Power Factor Penalties Tariff" ARAB CIGRE 2007 Regional Conference, Amman – Jordan 3-5 Sept, 2007 (accepted for presentation)

[4] Chafic Abi Said "Electric Energy & Energy Policy Program in Lebanon" GNESD Project 18/05/2005. March 31, 2005.

[5] Electrical Energy Tariff 2005, Effected from 1/1/2005 "Electricity Regulatory Commission Bulletin "Electrical Energy No. 56/2004 (391) date 17/12/2004 Warsaw - Poland,

[6] Ryan Firestone & Chris Marnay, Ernesto Orlando Lawrence, Berkeley National Laboratory, Distributed Energy Program, "The Effects of Tariff Structure on Distributed Generation Adoption in New York State" LBNL- 57942. Sept. 2005.

AUTHORS BIOGRAPHY

Ayman Agha He received M.Sc. and Ph.D. degrees in electrical engineering from the University of Mining and Metallurgy in Cracow, Poland in 1989 and 1997 respectively. He joined Jordan Phosphate Mines Co. (JPMC) in 1989. From 2000 to 2004 he worked as a Studies and Design Dept. Manger in alfanar Construction Co in Riyadh - KSA. In 2004 rejoined his previous work in (JPMC). His interests include power systems, nonlinear systems, optimization theory and reactive power compensation

Nabeel I. Tawalbeh He received his B.Sc. and Ph.D. degrees in electrical engineering from the University of Damascus- Syria and Warsaw University of Technology, Poland in 1985, 1997 respectively. Currently he serves an assistant Prof. at the Electrical Engineering Dept. at Al-Isra University-Jordan. His interests include planning, and design of distribution power systems, Load estimation, safety management, quality assurance and standardization.

MODELLING OF ENERGY MARKET PROCESSES IN THE DISTRIBUTOR MONOPOLIST STRUCTURE

Henryk Kaproń
Faculty of Electrical Engineering and Computer Science
Lublin University of Technology
Nadbystrzycka 38A, 20-618 Lublin, Poland.
E-mail: h.kapron@pollub.pl

Tomasz Kaproń
Pentacomp IT Systems
Lektykarska 29, 01-687 Warsaw, Poland.
E-mail: tok@pentacomp.pl

KEY WORDS
Model Analysis, Energy, Support

ABSTRACT

In this article the mathematical model for energy demand was presented as a function of its price, market share and price elasticity of demand. This model was used to determine analytically the extra profit for the energy distributor being the monopolist in relation to energy producers and consumers. Moreover, some examples of calculation of the extra profit possible to achieve by the distributor monopolist were presented.

INTRODUCTION

In order to ensure the correct functioning of the companies from the power sector in the market economy the companies from the power sector must aim at obtaining profit on their operation on the satisfying level. This task is not easy under market conditions and changing energy demand.

In case of the distribution of energy, the distributor being the monopolist in relation to energy producers and consumers has a chance to achieve additional profit as a result of market play, mainly on the production side. In this respect the market share of individual producers is important as well as their supply elasticity and ownership structure.

The interest in the possible change of the energy supplier in Poland is insignificant and in other countries there is not too much interest in it. The reason for this situation are well established habits of consumers as well as envisaged relatively insignificant benefits resulting from the change of the energy supplier, especially in case of small consumers. Predicting behaviour of energy consumers under market conditions is very important for energy producers and energy distributors. The behaviour of the energy distributor on the local market is of particular importance if it is the only supplier for many consumers who do not have a possibility of buying energy from many producers. Such a situation has been described in the present article.

MODELLING ENERGY DEMAND AND SUPPLY

The variability of energy demand depends mainly on the unit price of energy as well the time t, i.e.:

$$D = D(t, p)\,. \qquad (1)$$

The demand changes depending on the time, day, month, season of the year. However, taking into account the fact that the settlements between the producer and distributor for the energy consumed are made at determined time intervals, market prices can be expressed directly in the demand function. Mathematically, this situation is described in the best way by the following dependence:

$$D(p) = ap^{-\gamma}, \qquad (2)$$

where: D – energy demand, GJ, a – market value, EUR, p – energy price, EUR/GJ, γ - price elasticity of demand, -.

In the research on energy supply and demand a significant role is played by the price elasticity of demand. The elasticity of demand forces definite behaviour of producers and consequently their definite elasticity of supply. Under market conditions (as well as under other conditions) producers are obliged by signed agreements to execute energy supply. There are some technical limitations, however, as well as some economic requirements related chiefly to achieving rational prices.

By the producer rational price we should understand such a price that can be achieved in a given market structure and as a result of a definite legal regulations (tariff). The technical and economic requirement related to the energy distributor is minimising transmission losses and transmission costs. Thus, taking into account the technical aspects related to the source of energy and losses in the transmission network we can suggest, based on the paper [2] the rational heat price and supply formula in the definite market and legal environment in the following form:

$$p = \frac{\gamma}{(\gamma-1)(1-s)}\left(\frac{c_p}{w\eta} + k_n\right)$$

$$S = a\left[\frac{\gamma}{(\gamma-1)(1-s)}\left(\frac{c_p}{w\eta} + k_n\right)\right]^{-\gamma}, \qquad (3)$$

where: c_p and w are price and calorific value of the primary fuel, η - watt- hour efficiency, s – transmission losses , k_n – unit variable costs which do not depend on the type of primary fuel respectively.

The price elasticity of supply determines the ability of a dominant producer of energy to use its privileged position to determine market process. It expresses the relationship of relative changes of supply to relative changes of prices.

Taking into account the technical requirements of energy sources as well as the possibility of achieving satisfying income, the demand and consequently – the supply, depend-

ing on the rational price obtained on the market is fulfilled by the following inequality:

$$N \le D(p) \le M, \tag{4}$$

where: N, M mean load equal to technical minimum and power rating respectively.

MONOPOLIST DISTRIBUTOR ON THE LOCAL ENERGY MARKET

In the article the market structure has been analysed involving many competing producers who supply energy to one monopolist distributor who holds the privileged position both in relation to other producers (for whom it is a monopolist consumer) as well as to consumers (for whom it is a monopolist supplier). The following assumptions have been adopted:

- energy transmission takes place does not involve any losses,
- payment tariffs for energy, both purchased and sold, are one-component ones (there are no payments for power),
- prices for energy purchased by the distributor from various producers are the same (at the price c),
- price for energy sold by distributors to various consumers are the same (price p),
- distributor's costs are the sum of fixed costs and linear variable costs,
- energy demand depends only on its price p and has a constant price elasticity γ>1,
- energy supply depends only on its price c and has a constant price elasticity ε.

Taking into account the above assumptions the distributor's profit can be expressed by the formula:

$$\begin{aligned} Z(p,c) &= pD(p) - K_s - k_1 D(p) - cD(p) = \\ &= D(p)\left[p - k_1 - c\right] - K_s = \\ &= ap^{-\gamma}\left[p - k_1 - c\right] - K_s . \end{aligned} \tag{5}$$

Energy transmission without losses means that the supply and demand are equal

$$bc^{\varepsilon} = ap^{-\gamma}. \tag{6}$$

Thus after eliminating c

$$Z(p) = ap^{-\gamma}\left[p - k_1 - \alpha p^{-\beta}\right],$$

where

$$\alpha = \left(\frac{a}{b}\right)^{1/\phi}, \qquad \beta = \frac{\gamma}{\varepsilon}. \tag{7}$$

Equating the profit derivative to zero gives the price equation

$$p = \frac{\gamma}{\gamma - 1} k_1 + \alpha \frac{\gamma + \beta}{\gamma - 1} p^{-\beta}. \tag{8}$$

The explicit solution p^* of this equation in the set of positive numbers always exists and can be calculated using the method of consecutive approximations. Let us determine

$$\begin{aligned} F(p) &= \frac{\gamma}{\gamma - 1} k_1 + \alpha \frac{\gamma + \beta}{\gamma - 1} p^{-\beta} \\ p_0 &= \frac{\gamma}{\gamma - 1} k_1 \\ p_{n+1} &= F(p_n). \end{aligned} \tag{9}$$

In this method function $F(p)$ operates in the domain $p \in [p_0, \infty]$ and takes the value in the set $[p_0, F(p)]$. The maximum value of the function derivative in this domain amounts to

$$\begin{aligned} \left|F'(p_0)\right| &= \beta\alpha \frac{\gamma + \beta}{\gamma - 1} p_0^{-\beta - 1} = \\ &= \beta\alpha \frac{\gamma + \beta}{\gamma - 1} \left(\frac{\gamma}{\gamma - 1} k_1\right)^{-\beta - 1}. \end{aligned} \tag{10}$$

Thus the condition

$$\left|\frac{p_{n+1} - p_n}{p_n - p_{n-1}}\right| < 1 \tag{11}$$

which guarantees the concurrence ($p_n \to p^*$) of the method of consecutive approximations, takes the form

$$\alpha < \beta^{-1} (\gamma + \beta)^{-1} (\gamma - 1)^{-3\beta} \left(\gamma k_1\right)^{3\beta + 1}. \tag{12}$$

Let us notice that the rational sale price p^* implies the purchase price

$$c^* = \alpha (p^*)^{-\beta}. \tag{13}$$

According to [1] the rational price of the producer monopolist amounted to (we put the term of equation (8))

$$p^* = \frac{\gamma}{\gamma - 1} k_1. \tag{5.1}$$

(14)

Thus for the intermediary monopolist the sale price should be not more than

$$\frac{\gamma}{\gamma - 1} k_1 + c^* = \frac{\gamma}{\gamma - 1} k_1 + \alpha (p^*)^{-\beta}. \tag{15}$$

The privileged position of the monopolist on both sides of the market allows him to dictate the price that is even higher

$$\frac{\gamma}{\gamma - 1} k_1 + \alpha \frac{\gamma + \beta}{\gamma - 1} (p^*)^{-\beta}, \tag{16}$$

since

$$\frac{\gamma + \beta}{\gamma - 1} >> 1. \tag{17}$$

What is more, it is the price which allows the intermediary (operating as an individual business entity) to maximise his profit. Thus the price results from the economic account. This example shows how the defective market structure leads

in a natural way to overstated prices. The price difference (surplus) amounting to

$$\frac{\gamma}{\gamma-1}k_1+\alpha\frac{\gamma+\beta}{\gamma-1}(p^*)^{-\beta}-\frac{\gamma}{\gamma-1}k_1-\alpha(p^*)^{-\beta}= \\ =\alpha\frac{1+\beta}{\gamma-1}(p^*)^{-\beta}, \tag{18}$$

gives the extra profit to the double monopolist distributor amounting to

$$\text{extra profit} = \alpha\frac{1+\beta}{\gamma-1}(p^*)^{-\beta}. \tag{19}$$

CALCULATIONS RESULTS

Some examples of calculations of the extra profits achieved by the distributor monopolist have been presented in Fig. 1. As it results from these calculations the value of extra unit profits is strongly conditioned by the initial rational producer price and their price elasticity ε. The greater price elasticity results in the higher extra profit achieved by the distributor monopolist.

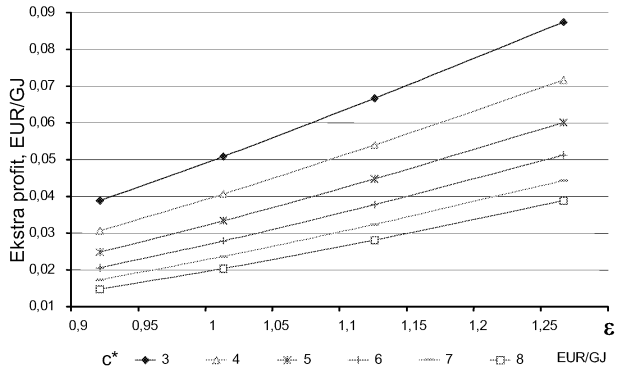


Fig. 1. Extra profit in the price elasticity function of supply for various producer prices

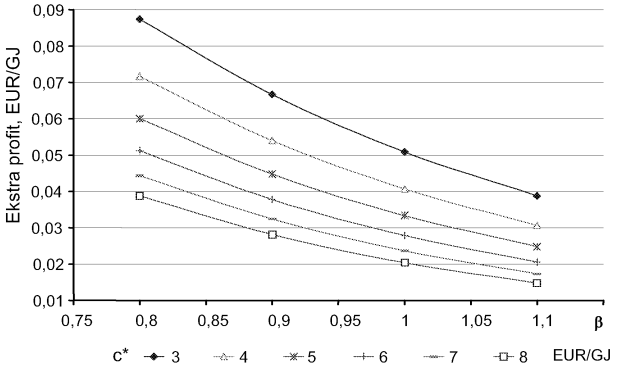


Fig. 2. Extra profit in the function of the demand elasticity to the supply elasticity ratio for various producer prices

CONCLUSIONS

From the above discussion it results that generating profit by the distributor monopolist depends chiefly on the market share of individual energy producers, their price elasticity and the amount of rational prices. In case of higher producer prices, the distributor monopolist using their marketing skills can generate higher extra profit.

In order to ensure the correct functioning of the energy market there should be a possibility of competition development both as regards producing energy and its distribution. In the free market environment, the choice of energy suppliers by energy consumers will be mainly conditioned by such criteria as price, reliability and quality of supplies. These criteria will force power companies to compress their business operating costs, to be more thrifty and to implement modern solutions as regards energy producing and transmission.

One can conclude then that there is a need for cooperation between the energy producer and distributor regarding sharing project costs and execution of distribution networks in order to consider economic arguments of consumers energy users.

REFERENCES

Kaproń, H. 2002. "The Efficiency of District Heat Generation in Market Conditions". Science Work of Environmental Engineering Faculty, No. 43. Warsaw Technical University Printing Office, Warsaw (December).

Kaproń, H. 2004. "Rational Supply and Price of Heat Within a District Heating System". Energy Market, No. 1, Kaprint Publishing Company (February).

Kotowicz, J. and Bełch, K. 2006."Comparative Analysis of Some Low Power Gas Turbines Cycles". Energy Market No. 2, Kaprint Publishing Company (April).

Sroczan, E. 2005. "The Simulation of Power System Structure Effects on Technical and Economic Effectiveness of Energy Generation". In: Modelling and Simulation. Ed.: Manuel Feliz-Teixeira J. and Carvalho-Brito A.E. Publ. EUROSIS-ETI. Ghent Belgium.

BIOGRAPHY

HENRYK KAPROŃ, Professor at the Faculty of Electrical Engineering and Computer Science Lublin University of Technology. From 1995, prof. Kaproń has been supporting the CHP Lublin-Wrotków Ltd. as a member of the operation control team. He is also Editor-in Chief of Energy Market Journal. Author of the book on energy conversion. He is a member of Polish Electricians Society and Vice President of Polish Group of International Association for Energy Economics.

TOMASZ KAPROŃ is employed as a software developer in Pentacomp IT Systems working for a mobile telephone network company project. Before, he worked for a few companies in Poland with responsibility for different software and hardware issues. He graduated in 1999 from Lublin University of Technology from the Faculty of Electrical Engineering, he also did postgraduate study at Warsaw School of Economics, organized in cooperation with Microsoft Poland. The topics of study were the analysis, design and implementation of software solutions to face the corporate needs.

ZERO REPLENISHMENT PHENOMENON IN A COLLABORATIVE FORECASTING PLANNING AND REPLENISHMENT ENVIRONMENT

Salvatore Cannella
Elena Ciancimino
Inter-Departmental Research Centre on Economic and Technology Planning (CIRPIET)
Faculty of Engineering, University of Palermo
Parco d'Orleans, 90128 Palermo, Italy
cannella@unipa.it, eciancimino@unipa.it

\

INTRODUCTION

Globalization, increased product variety, speeding up of technological innovation and shortening of product life cycles contributed to increase the complexity in configuring and managing intra-firm and inter-firm processes. Processes such as procurement, inbound logistics, internal operations and logistics, distribution and new product development involve now many different actors, inside and outside the company, and require several different skills: this scenario pushes companies to modify their supply chain strategy. Supply chain collaboration has been strongly advocated (Holweg et. al, 2004) since the mid 1990's under the banner of concepts such as Vendor Managed Inventory (VMI) and Collaborative Forecasting Planning and Replenishment (CPFR).

The amplification of the variance of order rate in multi-echelon systems, commonly known as 'demand amplification' or the 'bullwhip effect' (Lee et. al.) is a widely discussed topic in the field of supply chain management (Holweg and Disney, 2005). Bullwhip is an important measure, being symptomatic of a poorly performing supply chain. (Perona et. al, 2002)

As reported by Dejonckheere et al. (2004) the operation research community distinguishes three different solving approaches to supply chain instability: (1) redesigning the physical process, (2) redesigning the information patterns, (3) redesigning the decision process. In our work we conjointly adopt the second and the third approach through the application of IOPBCS replenishment rules on progressive sharing data environments.

The aim of this paper is to examine the effect of IOBPCS (Inventory and Order Based Production Control System) smoothing order policies and sharing information for variable signal demand in a four-tier supply chain. The ordering policies are based on exponential smoothing demand forecast. In continuous time domain approach TL-DE-APIOBPCS, SCDI-DE-APIOBPCS and VMI-DE-AVPVIOBPCS are investigated. Chen et. al. Order Rate Variance, Disney and Towill Inventory Variance and Zero-Replenishment Event are the measurement methods adopted.

SUPPLY CHAIN MODELLING

Most recent research methodologies on improving the dynamic behaviour of manufacturing businesses and supply chain may be categorized as management games, empirical studies, statistical, continuous and discrete control theory techniques, simulation and system dynamics (Disney et al., 2004). The last approach has the advantage of being able to model non-linearities whilst avoiding complicated mathematics.

In our examples we consider three different four-echelon supply chains characterized by a progressive information status on replenishment-production-delivery supply chain structure. The adopted order policies belong to the IOBPCS family (Towill, 1982; John et. al., 1994), which consists of a range of Production and Inventory Control (PIC) systems with five main components: a forecasting mechanism, a set of time values, an inventory feedback loop, a work in progress feedback loop and a target net stock setting. (Lalwani, 2006).

The three scenarios are modeled through smoothing order policies in which the aforementioned five components depend from the depth of information sharing. For every model a different typology of IOBPCS is used in order to emphasize the dynamics of the decision rule with enriched information (Mason-Jones and Towill, 1997). The subset of replenishment family rules is the DE-APIOBPCS studied by Deziel and Eilon (1967),

which refers to the case when $T_y^i = T_w^i$, that is equalizing the pipeline and inventory recovery times. The DE design lies well within the stable regime with extremely satisfying behaved dynamics response. For a extended guide to several guidelines of PIC smoothing replenishment rules components and optimum parameter setting in multi-echelon system see Disney and Towill (2003a, 2003b, 2006), and Boute et. al. (2007)

The impact of the smoothing order policies implemented in multi-stage systems shown in this paper is analyzed via continuous simulation (Forrester, 1961). Table 1 reports the operations management variables and parameters used in this work.

Material variables	
Wip_t^i	pipeline/work in progress at echelon *i* at time *t*
Inv_t^i	inventory of finished materials at echelon *I* at time *t*
S_t^i	units/orders finally shipped at echelon *i* at time *t*
F_t^i	throughput of work in progress at echelon *i* at time *t*
Information variables	
$\hat{d}_t^i$	tiers demand forecast at echelon *i* at time *t*
$\hat{d}_t^{cust}$	customer demand forecast at echelon *i* at time *t*
R_t^i	replenishment, order quantity at echelon *i* at time *t*
B_{t-1}^i	existing backlog of orders at echelon *i* at time *t*
$TInv_t^i$	target inventory at echelon *i* at time *t*
$TWip_t^i$	target work in progress at echelon *i* at time *t*
$VirtInv_t^i$	virtual inventory in VMI at echelon *i* at time *t*
$VirtWip_t^i$	virtual work in progress in VMI at echelon *i* at time *t*
$TVirtWip_t^i$	target virtual work in progress in VMI at echelon *i* at time *t*
$TVirtInv_t^i$	target virtual inventory in VMI at echelon *i* at time *t*
Parameters	
d_t^{cust}	customer demand at time t
α	forecast smoothing factor
T_p^i	physical production/distribution lead time at echelon *i* at time *t*
T_c^i	cover time for the inventory control at echelon *i* at time *t*
T_y^i	smoothing inventory parameter at echelon *i* at time *t* (describes of the discrepancy between actual inventory and target inventory levels)
T_w^i	smoothing work in progress parameter at echelon *i* at time *t* (describes of the discrepancy between actual WIP and target WIP levels)

Table 1. Models Nomenclature

ORDER POLICIES

The TL-DE-APIOBPCS (Traditional Linked - Deziel and Elion - Automatic Pipeline, Inventory and Order Based Production Control System) is tested in the first modeling configuration. Figure 1 illustrates the IOBPCS components and their relationships for a single tier.

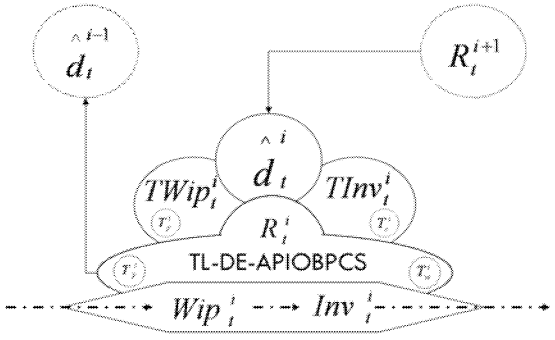


Figure 1. TL-DE-APIOBPCS information pattern

The SCDI-DE-APIOBPCS (Sharing Customer Demand Information - Deziel and Elion - Automatic Pipeline, Inventory and Order Based Production Control System) is the order policy implemented when market demand forecast, based on the end consumer order rate, and tiers' demand forecast at echelon *i,* based on the order placed by the subsequent stage, are simultaneously utilized (Fig. 2).

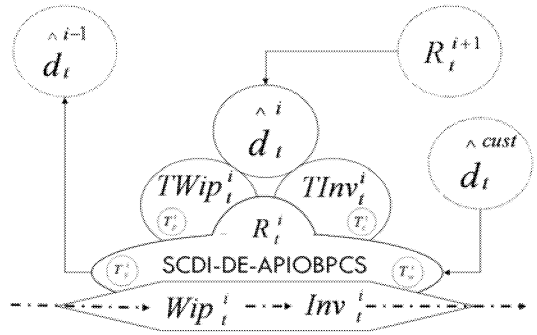


Figure 2. SCDI-DE-APIOBPCS information pattern

The VMI-DE-AVPVIOBPCS (Vendor Managed Inventory - Deziel and Elion - Automatic Virtual Pipeline, Virtual Inventory and Order Based Production Control System) is used in the last multi-echelon, and is based on market demand forecast data and on the integrated information on tiers' inventory and work in progress states (Fig. 3).

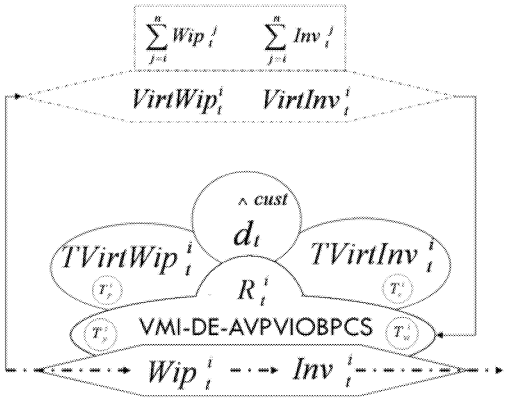


Figure 3. VMI-DE-APIOBPCS Information Pattern

In continuous time domain approach (Disney et. al., 2006) we focus on the amplification of demand signal. Difference equations are used to model progressively integrated Production Networks (Bruccoleri et. al., 2006). We adopt a modelling approach based on difference equations, solved through Vensim software with time step 0,25 over 52 simulation time units. The smoothing order policies strictly follow the order of event used in the Beer Game (Sterman, 1989). Table 2 includes the equations used to describe the patterns of materials and the demand forecast.

Good Flows	
$Wip_t^i = Wip_{t-1}^i + S_t^{i-1} - F_t^i$	(1)
$Inv_t^i = Inv_{t-1}^i + F_t^i - S_t^i$	(2)
$B_t^i = B_{t-1}^i + R_t^{i+1} - S_t^i$	(3)
$S_t^i = MIN(R_t^{i+1} + B_{t-1}^i; Inv_{t-1}^i + F_t^i)$	(4)
$F_t^i = S_{t-T_p}^{i-1}$	(5)
$R_t^i \geq 0, \forall i$	(6)
Demand forecast	
$\hat{d}_t^i = \alpha R_{t-1}^{i+1} + (1-\alpha)\hat{d}_{t-1}^i$, $0 < \alpha \leq 1, \forall i \neq 4$	(7)
$\hat{d}_t^{cust} = \alpha d_{t-1}^{cust} + (1-\alpha)\hat{d}_{t-1}^{cust}$; $0 < \alpha \leq 1$	(8)
$i = \{1,2,3,4\}$	

Table 2. Difference Equations describing material flow and demand forecast

Equations (7) and (8) represent the exponential smoothing formulae to forecast demand (Makridakis et. al, 1978). We adopt this distinction to take into account the forecast on customer order and the forecast on order placed by tiers. This formalism contributes to clarify the relationship between the enriched scenarios and the replenishment rules explored, in order to emphasize the correlated evolution of the five main IOBPCS components along the three multi-echelon schemes.

TRADITIONAL LINKED SUPPLY CHAIN MODEL

The replenishment order (9) is equal to the sum of the exponential demand forecast, plus the smoothed inventory difference between target stock and actual stock, plus the smoothed difference between target orders-in-the-pipeline and actual order placed but not yet received (Jonh et. al., 1994). The TL-DE-APIOBPCS is described by the following conditions:

$$R_t^i = \hat{d}_t^{\,i} + \frac{1}{T_w^i}(TWip_t^i - Wip_t^i) + \frac{1}{T_y^i}(TInv_t^i - Inv_t^i) \quad (9)$$

$$\hat{d}_t^{\,i} = \hat{d}_t^{\,cust} \Leftrightarrow i = 4 \quad (10)$$

Equation (10) shows that only the tiers next to the customer carry out a forecast based on market demand, while at the upstream stages the input demand data is given by equation (7).

The smoothed inventory and pipeline differences enable a dynamic recovery of safety stocks on replenishment-production-delivery multi-tier body. Target inventory (11) is updated every period according to the covering time and the new demand forecast. Target orders are seated in the pipeline as a depending variable by the demand forecast and production/delivery lead time (12).

$$TInv_t^i = \hat{d}_t^{\,i} T_c \quad (11)$$

$$TWip_t^i = \hat{d}_t^{\,i} T_p \quad (12)$$

SHARING CUSTOMER DEMAND INFORMATION MODEL

The order decision rule implemented in the second model (13) takes into account the conjoint use of (7) and (8). Customer demand forecast provides the first term of SCDI-DE-APIOBPCS, while Target Inventory (11) and Target Work in Progress (12) depend on tiers' demand forecast as well as in the TL model. The Inventory Control policies at every echelon depend on information incoming from the subsequent stage.

$$R_t^i = \hat{d}_t^{\,cust} + \frac{1}{T_w^i}(TWip_t^i - Wip_t^i) + \frac{1}{T_y^i}(TInv_t^i - Inv_t^i) \quad (13)$$

VENDOR MANAGED INVENTORY MODEL

VMI-DE-AVPVIOBPCS (16) is based on the concept of Virtual Inventory (VI) and Virtual Pipeline (VI). In the real business world the aforementioned Virtual Safety Stock can be associated to a Data Driven-DSS (Decision Support System) provided by a cyber infrastructure that releases continuous feedback on the multi-echelon material flow data. Herein VI and VP are constituted respectively by the sum of the Inventory at stage *i* and Inventories of subsequent echelons (14), and the sum of orders-in-the-pipeline at stage *i* and the work in progress of downstream tiers (15).

$$VirtInv_t^i = \sum_{j=i}^{4} Inv_t^j \quad (14)$$

$$VirtWip_t^i = \sum_{j=i}^{4} Wip_t^j \quad (15)$$

The smoothed difference terms in our VMI-IOBPCS typology consist of the gap recovery of a virtual safety stock at every echelon, integrating information available on inventory and work in progress values of the suppliers. Equations (16) and (17) represent the two variables used in the VMI model to harmonize replenishment patterns, in order to provide system scalability.

$$TVirtInv_t^i = \hat{d}_t^{\,cust} \sum_{j=i}^{4} T_c^j \quad (16)$$

$$TVirtWip_t^i = \hat{d}_t^{\,cust} \sum_{j=i}^{4} T_p^j \quad (17)$$

The VMI-DE-AVPVIOBPCS replenishment rule is formalized as follows:

$$R_t^i = \hat{d}_t^{\,cust} + \frac{1}{T_w^i}(TVirtWip_t^i - VirtWip_t^i) + \frac{1}{T_y^i}(TVirtInv_t^i - VirtInv_t^i) \quad (18)$$

RESULTS

In order to infer on the above described IOBPCS smoothing order policies, we adopt three assessments: the Order Rate Variance, the Inventory Variance and the Zero-Replenishment Event.

The Order Rate Variance measure (19) was proposed by Chen et al. (2000) to quantify the bullwhip effect for Order-up-to (OUT) policies

based on exponential smoothing forecast in an enriched supply chain.

$$Bullwhip = \frac{\sigma^2_{R^i} / \mu_{R^i}}{\sigma^2_{d^{cust}} / \mu_{d^{cust}}} \qquad (19)$$

$\sigma^2_{R^i}$ and μ_{R^i} describe the variance and the order quantity rate mean value at echelon *i*, $\sigma^2_{d^{cust}}$ and $\mu_{d^{cust}}$ denote the variance and the mean value of market demand.

Although very useful to quantify order rate variance, formula (19) does not include information on the stationary inventory variance, which is one of the causes of stockholding, poor customer service and supply chain volatility. Disney and Towill argue that Chen et. al. formula is only one-half of the bullwhip problem as the replenishment rule also influences the inventory dynamics (Hosoda, 2006). They present an additional measure of multi-echelon system instability, obtained by quantifying the inventory variance magnitude as the inventory variance and customer demand variance ratio, in order to explore the impact of the deleterious effect caused by demand amplification. At the same time, equation (20) offers additional value to investigate the replenishment-production-delivery system background. Suggested by the System Dynamics Logistic Group of the Cardiff Business School studies, we utilized Disney and Towill formula to quantify inventory stability.

$$VRInv = \frac{\sigma^2_{Inv^i} / \mu_{Inv^i}}{\sigma^2_{d^{cust}} / \mu_{d^{cust}}} \qquad (20)$$

Table 3 reports the Variance Ratio of Order, the Variance Ratio of Inventory and the input model parameters for each simulated environment. Fig. 4 plots Bullwhip Magnitude and Inventory Variance Ratio over the four echelon tiers.

	Order Rate Ratio		
Tiers	TL	SCDI	VMI
1	5,675	1,222	0,259
2	4,314	1,351	0,417
3	1,603	0,903	0,374
4	0,318	0,291	0,164

	Inventory Variance Ratio		
	TL	SCDI	VMI
1	161,6	44,7	12,4
2	248,7	58,6	13,8
3	145,4	55	15,9
4	30,2	31,6	21,3
$\alpha^i = 0{,}3$; $T_p^i = 2$; $T_c^i = 3$; $T_v^i = 3$; $T_w^i = 3$; $\forall i$			

Table 3. Chen et. al. bullwhip measure / Disney and Towill inventory variance

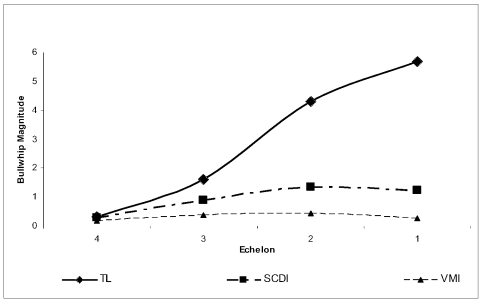


Figure 4. Impact of IOBPCS rules for variable signal demand

Figure 5 shows an interruption in the order flow for the TL and SCDI structures. This behavior is a direct consequence of demand amplification, due to a previous overly order rate. If condition (6) is not satisfied or the order rate has a null argument, the system faces what can be titled the Zero-Replenishment (ZR) phenomenon: a time lag in which at least one tier does not place any order, while market demand has reached a steady positive value. The ZR can be a criterion to evaluate the efficacy of the network decision policies for exogenous variation in a multi-echelon structure. ZR provides additional information to the instability measures available in operations research. In our work it is an assessment for the response delay of the system toward a variable demand signal input.

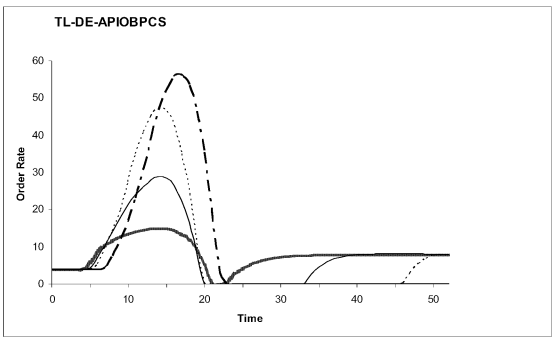


Figure 5. Zero-Replenishment phenomena in TL

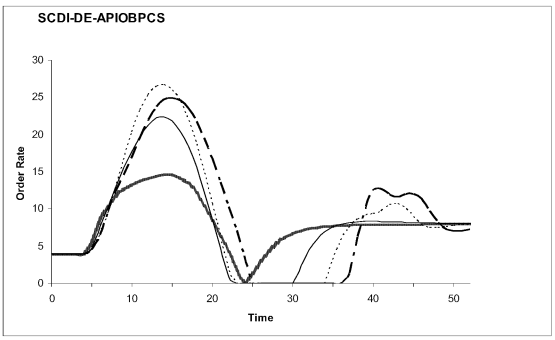


Figure 6. Zero-Replenishment phenomena in SCDI

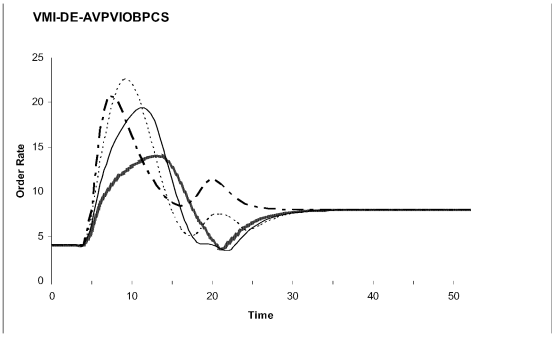


Figure 7. No-Zero-Replenishment phenomena in VMI

CONCLUSION

In this paper we explored a subset of IOBPCS under variable signal demand: TL-DE-APIOBPCS, SCDI-DE-APIOBPCS, VMI-DE-AVPVIOBPCS. To describe the efficacy of each policy we chose two measurement methods: Chen et. al. Bullwhip formula and Disney and Towill Inventory Variance Ratio. Furthermore we proposed as a performance indicator for order policies scalability the Zero Replenishment phenomenon.

Smoothing replenishment rules provide a solution to PIC problems through the reduction and avoidance of amplification of the variance of order rates in multi-echelon systems. In our models we wished to emphasize that IOBPCS efficacy is correlated to the depth of information sharing.

The TL-DE-APIOBPCS permits a considerable dampening of Bullwhip effect although the Traditional Linked has a structural tendency to demand amplification, summarized by Order-Inventory Ratios increasing monotonously along the four echelons and ZR phenomenon largely active in this chain. The demand amplification is avoided in SCDI-DE-APIOBPCS by placing in the order policy both customer demand forecast and tiers demand forecast, and also in VMI-DE-AVPVIOBPCS by virtual material stock data system. Although Bullwhip effect is dampened in both enriched supply chain, the response delay of the Production Networks for variable demand signal input is different. SCDI still exhibits the ZR phenomenon, while the VMI-DE-AVPVIOBPCS allows to achieve simultaneously demand amplification avoidance and ZR inhibition, providing multi-tier scalability.

References

Boute, R., N.; Disney, S., M.; Lambrecht, M., R.; Van Houdt, B.; (2007) An Integrated production and inventory model to dampen upstream demand variability in the supply chain. European Journal of Operational Reseach, Vol. 178, pp. 121-142.

Bruccoleri, M.; Perrone, G.; Cannella, S.; Crespo Marquez, A.; (2006). Production networks dynamics: analytical or non-analytical modelling?. Proceedings of the 14th International Working Seminar on Production Economics, Innsbruck, Austria, February 20-24.

Chen, F.; Drezner, Z; Ryan, JK., Simchi-Levi, D.; (2000). Quantifying the bullwhip effect in a simple Supply Chain: the impact of forecasting, lead-times and information. Management Science, Vol. 46 (3), pp. 436-43.

Deziel, D., P.; Eilon, S.; (1967) "A Linear Production – Inventory Control Rule." The Production Engineer, Vol. 43, pp. 93-104.

Dejonckheere, J.; Disney, S., M.; Lambrecht, M., R., Towill, D., R.; (2004). The impact of information enrichment on the bullwhip effect in Supply Chains: A control engineering perspective. European Journal of Operational Research, Vol. 153, pp. 727-750.

Disney, S., M.; Naim, M.,M.; Potter, A.; (2004). Assessing the impact of e-business on supply chain dynamics. International Journal of Production Economics, Vol. 89, pp. 109-118.

Disney, S., M.; Towill, D., R.; (2003a). The effect of vendor managed inventory (VMI) dynamics on the Bullwhip Effect in Supply Chains. International Journal of Production Economics, Vol. 85, pp. 199-215.

Disney, S.M.; Towill, D., R.; (2003b). On the bullwhip and inventory variance produced by an ordering policy. The International Journal of Management Science, 31 (3) 157-167.

Disney, S.; M.; Towill, D., R.; (2006). A methodology for benchmarking replenishment-induced bullwhip. Supply Chain Management: An International Journal, Vol. 11 (2), pp. 160-168.

Disney, S., M.; Towill, D., R.; Warburton, R., D., H.; (2006). On the equivalence of control theoretic, differential, and difference equation approaches to modeling supply chains. International Journal of Production Economics, Vol. 101 (1), pp. 194-208.

John, S.; Naim, M., M.; Towill, D., R.; (1994). Dynamic analysis of a WIP compensated decision support system. International Journal of Management Systems Design, Vol. 1 (4), pp. 283-297.

Forrester, J., W.; (1961). Industrial Dynamics. Sloan School of Management, MIT Press, Cambridge.

Holweg, M.; Disney S.M.; Holmstrom, J; Smaros, J.; (2005). Supply Chain Collaboration: Making Sense of the Strategy Continuum. European Management Journal Vol. 23,. (2), pp. 170–181.

Holweg, M.; Disney S.M.; (2005). The Evolving Frontiers of the Bullwhip Problem. Conference EurOMA: Operations and Global Competitiveness, Budapest Hungary June 19-22.

Hosoda, T.; Disney, S., M.; (2006). On variance amplification in a three-echelon supply chain with minimum mean square error forecasting. The International Journal of Management Science, Vol. 34, pp. 344-358.

Lalwani, C., S.; Disney, S., M.; Towill, D., R.; (2006). Observable and controllable state space representations of a generalized Order-Up-To policy. International Journal of Production Economics, Vol. 101 (1), pp. 173-184.

Makridakis, S.; Wheelwright, S., C.; McGee, V., E.; (1998). Forecasting. Methods and applications. John Wiley & Sons, West Sussex.

Mason-Jones, R.; Towill, D., R.; (1997). Information Enrichment: Designing the supply chain for competitive advantage. International Journal of Supply Chain Management, Vol. 2 (4), pp. 137-148.

Lee, H.,L; Padmanabhan, V.; Whang, S.; (1997). Information distortion in a Supply Chain: The Bullwhip effect. Management Science, Vol. 43 (4), pp. 546-558.

Perona, M.; Saccani, N.; (2004). Integration techniques in customer-supplier relationships: An empirical research in the Italian industry of household appliances. International Journal of Production Economics, Vol. 89, pp. 189-205.

Sterman J., D.; (1989). Modelling managerial behavior: misperceptions of feedback in a dynamic decision-making experiment. Management Science, Vol. 35 (3), pp. 321-339.

Towill, D., R.; (1982). Dynamic analysis of an inventory and order based production control system. International Journal of Production Research, Vol. 20, pp. 369-383.

SIMULATION SOFTWARE TOOLS

SIMULATION TOOLS

A SIMULATION STUDY OF AN AUTOMATED SYSTEM FOR A SHOE FACTORY USING ARENA

António Carvalho Brito
Faculdade de Engenharia
Universidade do Porto
Rua Dr. Roberto Frias; P-4200-465
Porto, PORTUGAL
E-mail: acbrito@fe.up.pt

Paulo Sá Marques
INESC-Porto
Campus da FEUP
Rua Dr. Roberto Frias, nº 378
Porto, PORTUGAL
E-mail: pmarques@inescporto.pt

KEYWORDS
Simulation, Modelling, Manufacturing, Arena.

ABSTRACT

One of the issues in the shoe industry is that market seeks the fulfilment of product orders for smaller production lots with shorter deadlines. A need therefore arises to respond to these market changes. Such a response includes amongst other things improving the internal organization of the enterprise, by for example rationalizing the production floor layout. A portuguese shoe factory decided to cut costs related to the transportation of all work in progress within the factory plant by implementing a fully automated storage and retrieval system.

This Paper presents the development of a simulation model, using the Arena simulation software tool, to analyse the performance of the automated storage and retrieval system based on a number of scenarios. The simulation model uses VBA (Visual Basic for Applications) to implement the control logic and graphics to visualize the sequences of movements and storage / retrieval operations.

1. INTRODUCTION

The production methodologies in the shoe industry were tradionally adapted to large production lots. As a result of the current market trend such methodologies are now obsolete. The physical distribution of equipment and production cells may also not be the most adequate for the quantity and diversity of the products that are produced. In addition to this, fixed costs associated with material and human resources may be reduced. Process optimization in the factories is therefore of vital importance specifically in regards to cutting costs, reducing production time and transportation time that occur between the various production centres. The process of finding better solutions to these problems are quite complex and simulation models can be extremely helpful.

Computer simulation has become widely used in the analysis of complex problems and is now an important tool in evaluation production and storage/retrieval systems.

In 1976 the concept of Visual Interactive Simulation (VIS) was introduced by Hurrion (Hurrion 1976). With VIS a image of the model running is displayed on the screen and the user can interrupt the model running, at any time, and interact with it in a way that he can influence the future behaviour of the model.

Visual Interactive Simulation aggregates a set of graphics, visual and interactive techniques which can be incorporated with different levels of extension in a simulation system. VIS helps the analyst in visually identify the problems and in finding solutions. In the past years, there have been a large number of situations where VIS was successfully applied to different areas, including manufacturing and warehousing (Brito and Basto 1992), (Dangelmaier and Bachers 1986), (Marín et al. 1998), (Feliz-Teixeira and Brito 2003), (Burnett and LeBaron 2001) and (Macro and Salmi 2002). Most of the current simulation software incorporate interactive and graphical facilities. Simulation software has modules such as process, stations, transporters, conveyors, etc., to help design and analyse manufacturing systems. A survey can be found in (Smith 2003).

That is the case of Arena (by Rockwell Software) that was used to developed a simulation system to evaluate the performance of the automated storage and retrieval system of a shoe factory and to test different control rules before they are implemented in the physical system.

2. THE FACTORY PLANT

A portuguese shoe factory installed a fully automated storage and retrieval system to allow the movement of containers between production cells. The system that was developed and installed in the factory consists of two subsystems, an automated transportation system and an automated warehouse. Together they move and store both work in progress and material in an aggregated fashion so that these can be used in the various shop-floor sections.

Two main sections were considered: sew and assembly. The automated warehouse itself consists of four storage and retrieval facilities (see Figure 1). The automated transportation system on the other hand consists of a vast array of conveyors and elevation towers that make the

interface between the warehouses and production units. The access points to these productions units are represented as red circles in Figure 1.

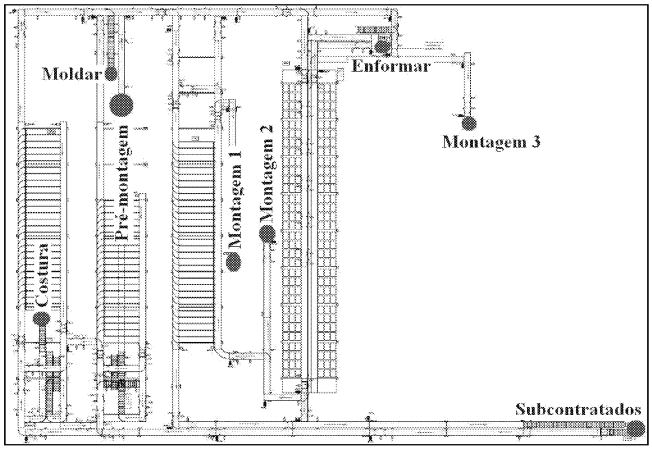


Figure 1: Factory Plant

The automation system is controlled by three PLCs (Programable Logic Controllers). The production management software runs on a server and communicates with the computer clients software and with the PLCs through a TCP/IP network (client/server). A computer client allows each production unit to access all the necessary information about the production plan. Each unit can follow the plan schedule by the ERP (Enterprise Software planning) or change it, if needed.

The materials and the Work In Progress (WIP) are transported in containers (see Figure 2). Each container can take several pair of shoes depending on the model size.

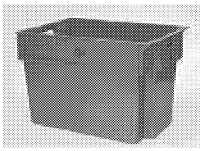


Figure 2: Container Used for Transporting Materials

The conveyors move the containers, with the Work In Progress (WIP) and materials, from the warehouses to the production units, back to the warehouses and between warehouses. The conveyor's speed is constant and equal to 1m/s. Pneumatic gates control the container's path at the conveyor's junctions.

The conveyor system works at a height of 3 meters and Input/Output towers are used as an interface with the production units at the ground floor (see Figure 3). Depending of the production units the I/O towers can have one or two buffers. One buffer (input) is used to keep the containers waiting at the production unit. The second buffer (output) is used to keep the containers waiting to be transported back to the storage system.

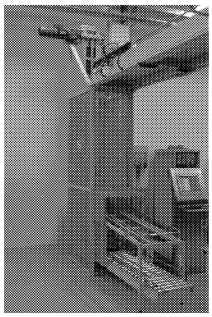


Figure 3: Input/Output Tower

There are 4 storage areas (see Figure 1) that are placed at a height of 3 meters and are connected by the conveyor system. There are two types of storage: the Live Storage and the robotized storage. In the Live Storage the containers flow by gravity in a bay that can store a maximum of 4 containers (see Figure 4). The three storage areas on the left of Figure 1 are Live Storage.



Figure 4: Live Storage

In the robotized storage a robot arm moves the containers from the conveyors to the storage area and in the reverse direction (see Figure 5).

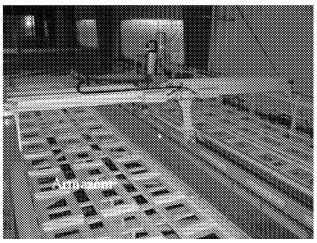


Figure 5: Robotise Storage

The Live Storage access logic can be quite complex, such as when a requested container is not at the first position in the bay, the system is forced to take out one or more containers to access the one required (see Figure 6).

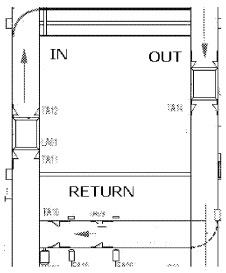


Figure 6: Live Storage Access Logic

The company needs to simultaneously manage a large number of different production routes. The performance exhibited by the automated storage and retrieval system depends on a number of variables that parameterize these production routes. The evaluation of the system performance and control rules is quite difficult as cannot be tested without disturbing the normal work of the factory. This situation lead to the decision of developing a simulation model where several solutions can be tested before being implemented.

3. THE SIMULATION MODEL

The simulation model development was decided in order to evaluate the performance and the control rules of the automated system. Some of the reasons to choose the Arena software were its graphical interface and the availability of several production modules (stations, conveyors, etc.) that were needed to model the system. Another important reason to choose Arena was the need to implement the system logic that is quite complex and specific from this industrial unit and that could only be implement through the use of a programming language, such as Arena VBA (Visual Basic for Applications).

The amount of data associated with the industrial system doesn't allow data to be inputted manually. The VBA is also used to read, define and generate internally many of the simulation data and elements. The data is read from Excel spreadsheets or directly from the industrial main SQL database. VBA modules are also used when the model logic is complex as in the containers data definition. The containers are boxes (see Figure 2) used to transport materials and the Work In Progress (WIP).

The containers (entities) are created in a Create module and its attributes created in the Assign module (See Figure 7). Some of the container attributes are the capacity (nº of pairs), the operation's sequence and the operation's time. In the VBA modules simulation data is defined dynamically and initial configuration parameters are set. In the first VBA module container attributes are defined and containers can be placed at storage area and in production units to prevent simulation start from an empty plant situation. In the second VBA module the container production plan attribute is defined and the container next operation attribute is set.

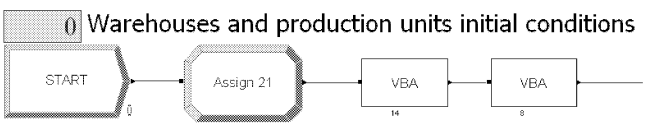


Figure 7: Container Initialization VBA Modules

When the simulation starts the user can define some of the model parameters such as the number of active input/output towers, number of containers or warehouse fill percentage (see Figure 8).

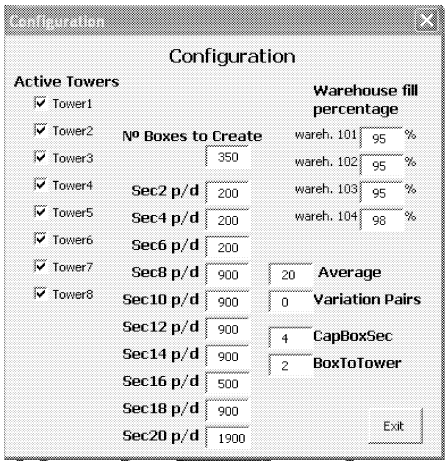


Figure 8: Initial Configuration Screen

A VBA module reads the configuration files. These data can be imported from the production database or generated internally and it is used to configure the warehouses, the production units and the input/output towers. Examples of warehouses configuration parameters are the storage type, the number of bays or the number of containers in each bay. Examples of production units configuration parameters are the capacity, the I/O access tower or the associated warehouse. Examples of I/O towers configuration parameters are the size (nº of containers) of the input or output buffers.

The model logic uses standard Arena modules, whenever it is possible, but most often it is required to use VBA modules to implement the system logic. That is the case of the I/O towers (see Figure 9).

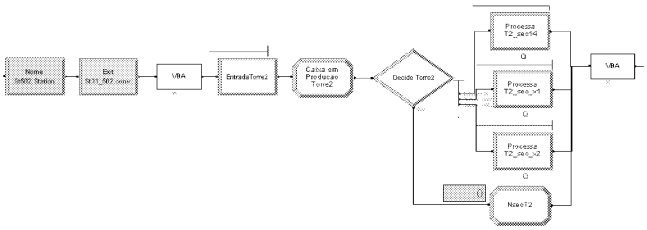


Figure 9: I/O Towers Model Logic

The first module is a Station used for animation purposes, followed by an Exit module that frees the conveyor cell as the container leaves the conveyor and enters the tower. The VBA module updates the orders list, check for new orders, updates container attributes, etc. One restriction that is taked care by VBA is related with the number of containers coming to the tower that can be simultaneously in the conveyor system. That is necessary to prevent input buffer overflow at the tower. A Hold module is used to keep the container in the buffer until it is required by the production unit. As one tower can supply more than one production unit the decision is made by a Decide module. The last VBA module defines the container next destination, based on the operations sequence, updates container attributes, etc.

The conveyor system installed at the plant is also quite complex mainly due to several types of junctions. In the model were included the following junction types: 1I1O, 2I1, 1I2, NI1O, NI2O and cross-junction. The codification used was the number of input conveyors followed by the letter "I" and the number of output conveyors followed by the letter "O". So, NI2O means several input conveyors and 2 output conveyors (see Figure 10).

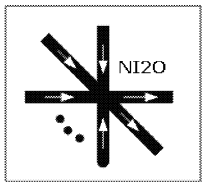


Figure 10: NI2O Conveyor Junction Type

The logic for this type of junction is showed in Figure 11. Similar blocks were developed for the several types of junctions.

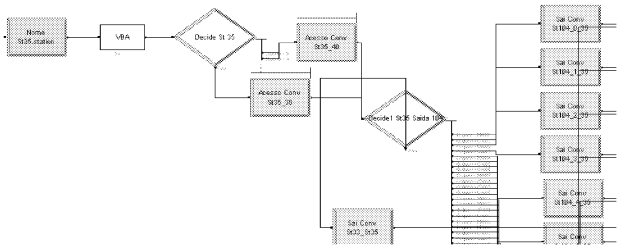


Figure 11: NI2O Conveyor Junction Type Logic

During the simulation run a graphical display is used to represent the dynamic of the model. A dxf drawing is used to give a realist representation of the system. The animated display shows the containers movement (conveyors), the warehouses and sections activity (see Figure 12).

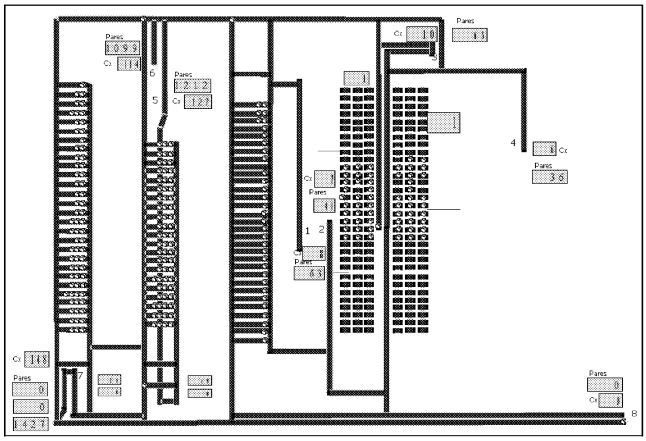


Figure 12: Simulation Display

A Convey module is use for the animation and Conveyor and Segment modules are used for the network data definition. The Convey module allows a container to be transported between stations. The Conveyor module defines the type and conveyor attributes and in the Segment module the conveyor path layout and distances are defined. The stations and segments are represented in Figure 13.

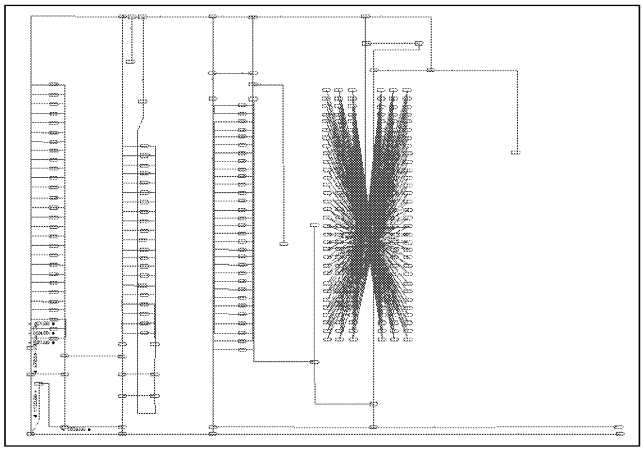


Figure 13: Stations and Segments Layout

Tests were done during the model development for verification purposes. The model validation was done by comparing the simulation results with those obtained from the factory plant. Also experts from the factory plant were involved to identify problems. Sensitivity analysis was used comparing the effect of small changes in input data on the simulation results. Replications were used to analyse the variability of the results. Animation was also an important tool during the validation process.

4. RESULTS

Results were obtained for different user scenarios. The scenarios were defined using different values for the input variables and simulation parameters. Examples of simulation parameters used are the length of the simulation run, the number of replications and the warm up period. Typical values for these parameters were 1 day for the simulation length, 1, 5 and 10 for the number of replications and 1 hour for the warm up period.

Examples of input variables are the containers capacity (nº of pairs), the number of pairs produced in each production unit and the products operations sequence. Table 1 shows a tipical daily production for the number of pairs in each production unit (tower).

Table 1: Number of Pairs in Each Tower

T1	T2	T3	T4	T5	T6	T7	T8
1000	1000	1000	500	2500	1000	200	2500

The use of different seeds in random number generation and the use of average and standard deviation values for the input variables allows statistical analysis of the final results. Arena reports gives information about minimum, maximum, average and half-width values for the output variables. Arena uses 95% confidence intervals by default. Figure 14 shows the average number of containers processed in the different production units (towers) in one day.

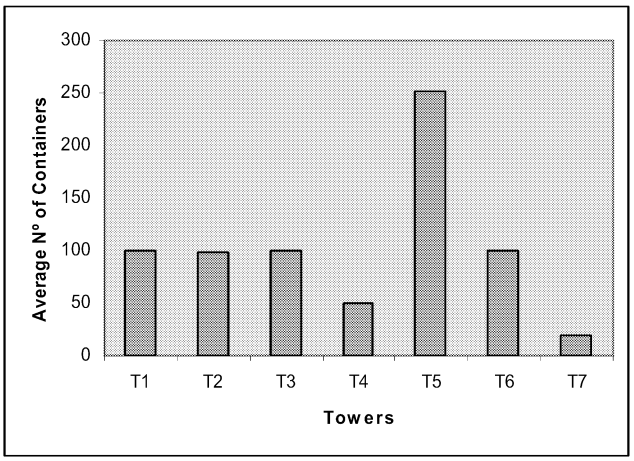


Figure 14: Average Nº Of Containers Processed / Tower

Process Analyzer, a Arena professional version tool, was used to compare different scenarios. Figure 15 presents the output obtained for the production unit (tower 7) utilization when producing 2500, 5000 and 5752 pairs of shoes.

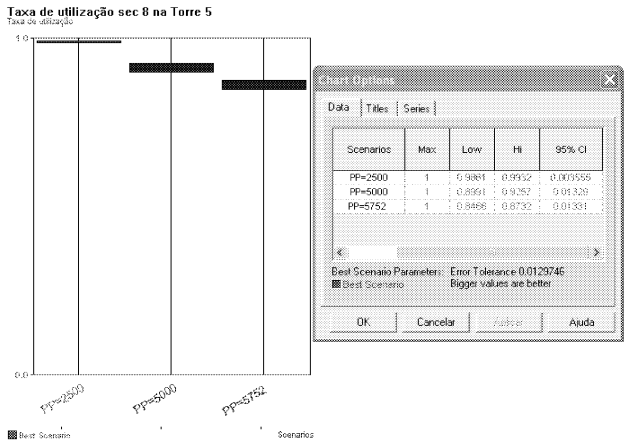


Figure 15: Production Unit (tower 7) Utilization

These results shows that the automated storage and retrieval system is not capable of supply the production unit when the production are increased above a certain level.

5. CONCLUSIONS

The use of simulation packages, such as Arena, can be very helpful in analyzing and improving production systems. The availability of modules such as process, stations, transporters, etc. turn easier the development of the model. However, there are systems, as the one presented, where the use of a programming language, like VBA, is necessary to model the complex logic. The use of colors, graphics and animation helped to validate the model and see why problems arise during its running and interpreting the simulation results. The simulation project objectives were achieved. The model showed that the automated storage and retrieval system is capable of supply the production units without delays and to cope with the expected demand increase in the near future. Different scenarios were analyzed helping to improve the production system. New rules for the control software could be tested, and used later, without disturbing the normal work at the factory. As future work, is expected to use VBA programming to automatically generate simulation models for new installations of similar type. This will allow a faster development of the models and also allow testing the control software before being installed.

REFERENCES

Brito, A. C. and J. A. Basto. 1992, "The Use of Computer Aided Design Techniques in Configuring Visual Interactive Simulation Models for Warehouse Design", Revue des Systèmes de Décision, v 1 , n 2-3, p 191 - 212

Burnett, D. and T. LeBaron, 2001, "Efficiently modeling warehouse systems" in Proceedings of the 2001 Winter Simulation Conference (Arlington, VA, Dec 9-12), IEEE, Picataway, NJ, p 1001-1006

Dangelmaier, W. and R. Bachers, 1986 "SIMULAP - A Simulation System For Material Flow and Warehouse Design", Material Flow, n 3, p 207-214.

Feliz-Teixeira, J. M., & Brito, A. E. S. C. (2003). An Approach for Dynamic Supply Chain Modelling. Paper presented at the 2003 European Simulation and Modelling Conference, Naples, Italy.

Hurrion, R., "The Design, Use and Requirements of an Interactive Visual Computer Simulation Language to Explore Production Planning Problems", Ph.D. thesis, University of London, 1976.

Macro, J. and R. Salmi, 2002, "A simulation tool to determine warehouse efficiencies and storage allocations" in Proceedings of the 2002 Winter Simulation Conference (San Diego, CA, Dec 8-11), IEEE, Picataway, NJ, p 1274-1281

Marín, R.M.; J. Garrido, J.L. Trillo, J. Saez, J. Armesto. 1998 "Design and simulation of an industrial automated overhead warehouse" Integrated Manufacturing Systems , v 9, n 5-6, p 308-313.

Smith, J.S. 2003. Survey on the use of simulation for manufacturing system design and operation. Journal of Manufacturing Systems 22 (2): 157.

AUTHOR BIOGRAPHY

ANTÓNIO CARVALHO BRITO was born in Porto in 1958, and went to Universidade do Porto, where he studied mechanical engineering and obtained his degree in 1981, staying in the same institution as Lecturer where he did his M.S. thesis in 1985. He obtained his Ph.D. in Simulation from the School of Management at the University of Cranfield (U.K.) in 1993. He is Auxiliar Professor at Universidade do Porto since 1993, where he leads research in the field of Visual Interactive Simulation. His e-mail address is: acbrito@fe.up.pt

PAULO SÁ MARQUES was born in Porto (Portugal) in 1963, and went to Universidade do Porto, where he studied electrical engineering, and obtained his degree in 1988, He is in INESC Porto since 1998, where he is now project manager. His e-mail address is: pmarques@inescporto.pt

VisualRTJSim - A SIMULATION TOOL FOR REAL-TIME JAVA

Werner Pohlmann and Markus Rabenberger
Department of Computer Sciences
University of Salzburg
A 5020 Salzburg, Austria
pohlmann@cosy.sbg.ac.at, mraben@cosy.sbg.ac.at

KEYWORDS
Real-Time Systems, Java, Simulation

ABSTRACT

RTJSim is a Java-based discrete-event simulator for real-time and embedded systems. It can take real-time Java programs (compliant with RTSJ, the Real Time Specification for Java) as input and execute them in simulated time on a standard Java VM. RTJSim is useful for testing and other kinds of execution-based studies on RTSJ programs. A first version of the simulator was presented at ESM05; here we focus on an extension, VisualRTJSim, that facilitates use and enables new kinds of experiments.

INTRODUCTION

Rigorous static analysis (e.g. verification, schedulability analysis) is indispensible for real-time systems, esp. if they are safety critical. Nevertheless, there is a demand for execution-based studies (e.g. design validation, software test and debugging, measurements of analytically unavailable figures). And simulation is very common in the real-time/embedded system domain, because it is useful at various stages of a project and can be done on everybody's desktop. System development environments tend to offer simulation services at a rather high-level of description. In contrast, our aim is to enable a less restricted approach: the simulator can take the completed real-time program as an executable model! This does not preclude that the developer starts simulative works with mere skeleton programs which will be fleshed out later. But it means that all modelling is done in the actual working language and not in some separate system description or simulation language which may be only loosely related to the developed software.

We choose Java for our undertaking. There are two reasons. First, Java is object-oriented, portable, easily available and widely-used. Second, the Real Time Specification for Java (RTSJ for short, several versions since 2002, cf. http://www.rtsj.org, cf. (Bollella et al. 2001), cf. (Dibble and Wellings 2004)) made it one of the very few high level programming languages in the area of real-time/embedded computing. Based on threads, it is comparable to Ada but in some respects more expressive and complete; see (Wellings 2003). RTSJ provides solutions for two key problems that arise when Java is employed for real-time programming. First, Java's thread model is insufficiently defined w.r.t. scheduling and interaction, which is a cause of unpredictable behaviour and thus intolerable in real-time programming. Second, Java's reliance on garbage collection leads to delays that in general are unpredictable too and therefore must be ruled out.

We called our simulator RTJSim because Java is our modelling as well as implementation language. A first presentation of the idea of RTJSim and the essentials of its realisation was given at ESM05 (Rosenkranz and Pohlmann 05). In the present paper, we explain several additions that are vital for the practical use of our framework that we now call VisualRTJSim. - Our paper is structured as follows: In the next section we briefly recall how our simulator works. The following sections cover new material: first, how the the original RTSJ program is preprocessed to be executable by the simulator, and second, how the experimentator can interact with the running simulation whose visualisation he follows. We then illustrate this material by a simple yet meaningful example. The final section sums up and describes areas for future work.

SIMULATED TIME AND WALL-CLOCK TIME

Using a real-time program as a simulation model of itself, we have to juggle with three notions of time: first, the physical time for which the program was written; second, the virtual (or simulated) time that depicts the first; third, the physical time (or wall-clock time) in which the simulation like any computation runs. In discrete-event simulation, virtual time is realised by a clock variable whose successive states derive from a queue of future-event notices. This technique, which we employ, associates virtual time with computational actions and does not bear any numerical relation to wall-clock time.

In RTJSim, virtual time advances to some definite future value for one of two reasons: either there is nothing to do now but something to do later in the simulated real-time system, or there is something going on now that will be finished later. The RTSJ `sleep(long d)`-method and a variant like `waitForNextPeriod()` exemplify the first alternative and simply translate into the insertion of a corresponding event notice into the event queue. More difficult is the case of time spent on some piece of computational work. The same work is done in the simulation, but its duration `d` has to be known or estimated for the target platform and charged against the simulation clock by additionally calling a method `consume(RelativeTime d)` that our framework defines for this purpose. The parameter for the call must give net computation time (typical: worst case execution

time, WCET) and not response time; the simulator is responsible to account for any delays caused by preemption.

The given real-time program consists of threads and asynchronous event handlers, the other kind of schedulable objects; for brevity let us here concentrate on threads (which are used for the event handling anyway). In analogy to to our distinction of virtual and wall-clock time, we must not mix up simulated concurrency and concurrency in the simulating system. It is a conceptual and possibly serious and hard-to-find practical error to mix them up. We decided to disallow concurrent activity in the simulating system and arrange matters so that at any point of wall-clock time all threads but one are stuck on a `wait()`. The main reason is this: RTSJ, in contrast to normal Java, requires a statically analysable scheduling strategy like fixed-priority scheduling (for the processor and all monitor-related queues) in conjunction with the priority inheritance protocol (to prevent unbounded priority inversion). And since we want to enable simulations on arbitrary JVMs, we have to explicitly program all the scheduling machinery (simulated processor contention, monitors ..) into the simulator. This gives us perfect control and has the nice effect of making simulation runs reproducible, but comes at a price: to prevent the JVM from directly executing monitor-related code of the given real-time program, we must replace all `wait()`, `notify()` etc calls by substitute methods defined in the simulator framework.

PREPROCESSING THE REAL-TIME PROGRAM

So we must alter the given real-time program in two ways: we must (1) insert calls of the consume-method that makes computations have a footprint in virtual time, and we must (2) re-interprete monitor-related code so that RTSJ scheduling becomes effective in the simulator.

For problem (1), the insertion of consume-calls, the obvious solution is to put suitable comments into the original program and then translate these annotations for simulation. (Due to classic undecidability results, completely automatic timing of program code is possible only for the most primitive, e.g. loop-free, case.) We employ special comments of the form

```
//{{consume(0.0023);
   .... //the computation
//}}
```

which translates into:

```
//{{
   consume(0.0023);
   ....  //the computation
//}}
```

The block-like form has no technical meaning (at present), but shall merely delimit the covered piece of code and act as a reminder to the modeller that fixing a duration is hardly compatible with side effects, indefinite blocking, premature abortion etc.; we may develop some rough sanity checks on this later. The foremost problem with consume calls is how to choose their parameters. At an early stage of development, even before program details are worked out, some plausible guess may be useful just to reserve some share of processor time for a subproblem. Later on, one needs realistic estimates, single (best/average/worst-case) values or random numbers drawn from a suitable distribution. The problem of obtaining such information has much in common with WCET analysis for static schedulability proofs. Recent research in this area (Schoeberl and Pederson 2006; Bernat, Burns and Wellings 2000) starts with the Byte Code (Java's machine language) form of the program, associates this with platform-dependent execution times and then re-formats the code into higher level structures (loops, branching statements) relying on annotations of the original program (including upper bounds for the number of iterations...). For our purpose, a reduced version of this recipe suffices: Since we do not need overall maximal values for use in static analysis, but simulate, we are happy to execute a control structure like a loop if only the duration of the body is given.

As regards problem (2), the re-interpretation of synchronisation-related syntax, let us briefly consider what the underlying problem is: Java's critical regions, marked by the keyword "`synchronized`", as well as the semantics of "`wait()`" and "`notify()`", rely on the lock that Java implicitly associates with every object. We cannot directly use this lock, and we cannot re-interprete wait or notify via subclassing. So to conform to the scheduling requirements of RTSJ, we have to build our own mechanism, which relies on a special synchronisation object that, as a scheduler service, can be created for every object. And we have to do a piece of translation as follows:

```
synchronized void aMethod(){....}
```

expands into

```
void aMethod(){synchronized(this){....}}
```

and then the general scheme is that

```
synchronized(o){
   ...
   wait();
   ...
   notify();
   ...
}
```

gives

```
RealtimeThread._enter(o);
try{
    ...
    RealtimeThread. _wait(o);
    ...
    RealtimeThread._notify(o);
    ...
```

```
}finally {RealtimeThread._leave(o); }
```

where all the new methods whose name begins with an underscore are defined as static in our version of class `RealtimeThread`, and their bodies start with requesting the synchronisation object for object o from the scheduler.

In contrast to the simple find&replace text processing that is necessary for problem (1), the transformation necessary for problem (2) implies that the preprocessor can handle nested structures to insert the correct object reference into the produced code. So we built a simple recursive-descent parser that drives the translation.

OBSERVATIONS AND EXPERIMENTS

Real-time programs are built to monitor or control some technical or natural system. So a simulated execution of such a program needs to represent the physical context by a piece of software that likewise evolves in virtual time. Since the focus is on the control program, the context model may be simple. The interface is a set of sensor and actuator variables. We cannot employ real-time threads for this additional modelling purpose because such threads are subject to scheduling. So we defined a class `SimulationEvent`, objects of which have a method `occurred()` that encodes what is achieved by the event in question, i.e. some durationless action. SimulationEvent objects are put on the event list of the simulator and executed by the threads moving forward in virtual time. Such events may be singletons or form chains, with one event generating the next for some future time.

As usual, a simulation run may write a stream of data to a file for later inspection or print a report from aggregate data. This may be useful for test runs, but for questions of understanding the dynamics of certain situations an interactive mode is desirable. At present we can offer three such possibilities for RTJSim. First, the user can step through the simulation run, i.e. have execution only move on when prompted. Second, the user can delay the progress of a real-time thread by interactively increasing the parameter of a consume-call; this is useful e.g. when looking for slacktime or when studying the consequences of possible cost overruns. Third, the user can throw in a new simulation event, i.e. act as part of the environment of the running program.

Let us briefly consider the technique behind such services. In RTJSim, there is no supervisor thread, but the real-time threads, after start, manage themselves and to this end make use of methods and information kept in the scheduler object and in the clock with its event queue. The active thread, when blocking, will find and release the next thing to happen, a thread to resume or an event to occur. In the interactive mode, the active thread will additionally ask for user input. In the case of interactive event insertion, we use Java reflection based on an event object prefabricated in the main method; the user is free to choose the parameters of the event (provided that occurrence time is in the virtual future).

All simulator output is based on logging observation events, which can be chosen from a rich predefined set. (Do not confuse observation events, simulation events and Java's asynchronous events.) We use log4j (cf. http://logging.apache.org/log4j), a tool that is in widespread use for debug purposes. This writes into a textfile; for interactive use, we offer a graphical presentation that, for the various threads, depicts observation events against the time axis.

As a general technical basis for our work we use Eclipse Rich Client Platform (RCP), see (http://www.eclipse.org). This is the platform that was extracted from the Eclipse IDE. Now one can use the same platform to build commercial quality applications with powerful user interfaces. VisualRTJSimTool is our name for RTJSim with the preprocessing and the interactive simulation added and realised as two Eclipse RCP perspectives. - The appendix shows two screenshots to exemplify how the new tool presents itself to the user.

AN EXAMPLE

For an illustration, we consider a small and simplified problem that may be part of the control of an elevator, a conveyor belt, a robot arm and the like. The elevator shall move from a given position to a destination position. This motion, which we simplify to be uniform, is monitored by a thread `MotionControl`, which reads a sensor variable `double position` and sets an actuator variable `boolean go` as long as the destination position is not reached. A second thread, `EmergencyControl`, prematurely ends the motion in reaction to a sensor variable `boolean stop` changing to true. `EmergencyControl` acts on `MotionControl` by calling the `finish()` method of the latter; sensor and actuator variables, together with getter and setter methods, are defined in a class `ElevatorInterface`. Here are the two threads, somewhat abbreviated and not-translated for RTJSim:

```
public class MotionControl
  extends RealtimeThread implements Finishable
{
  .....   //local definitions, e.g. period
  .....   //constructor

  public synchronized void finish() {
          closedown = true; }
  public synchronized Boolean finished() {
          return closedown; }

  public void run() {
    double pos = myInterface.getPosition();
    double endPos = myInterface.getEndPosition();
    while (!finished() && pos < endPos) {
       //{{RealtimeThread.consume(4);
       //}}
       myInterface.setGo(true);
       try {waitForNextPeriod();}
```

```
            catch
            (IllegalThreadStateException e) {...}
            catch (InterruptedException e) {...}
         pos = myInterface.getPosition();
      }
      myInterface.setGo(false);
   }
}

public class EmergencyControl
   extends RealtimeThread
{
  .....   // local definitions, e.g. period
  .....   //constructor

   public void run() {
      boolean gotSignal = myInterface.getStop();
      while (!gotSignal) {
         //{{RealtimeThread.consume(PERIOD - 3);
         //}}
         try {waitForNextPeriod();}
          catch
             (IllegalThreadStateException e){...}
         catch (InterruptedException e){...}
         gotSignal = myInterface.getStop();
      }
      otherThread.finish();
   }
}
```

The environment is modelled by two simulation event classes. The first, `PositionUpdateEvent`, models the spatial progress of the elevator, i.e. regularly calculates the new position and writes it to the sensor to be read by the MotionControl-thread. The second event class, `StopEvent`, models a stop signal which is interactively given by the user. It therefore has no provisions for iterated occurrence, whereas the position-update event re-inserts itself into the event queue to occur again and again until the actuator variable go is set to false. Further notice that events never call the consume-method because they do not belong to the real-time program (and so do not spend processor time) and are instantaneous anyway.

```
public class PositionUpdateEvent
      implements SimulationEvent
{
  .....   // local definitions
  .....   //constructor

  public void occurred (SimulationClock clock,
                        AbsoluteTime now) {
     //compute new position and write sensor
     int usedSpeed = SPEED;
     myInterface.getPosition();
     long delta =(updateInterval/1000)*usedSpeed;
     myInterface.setPosition
            (myInterface.getPosition() + delta );
     //prepare next occurrence
     if (myInterface.getGo()){
        AbsoluteTime next =
                    now.add(updateInterval, 0);
        clock.insertEvent(this, next);}
    }
}
```

```
public class StopEvent
      implements SimulationEvent
{
  .....   // local definitions
  .....   //constructor

  public void occurred (SimulationClock clock,
                        AbsoluteTime now) {
     // set stop sensor to true
     myInterface.setStop();
  }
}
```

At simulation start, the two thread classes are instantiated and the thread objects are started. Similarly, the two event classes are instantiated; but while the StopEvent object is kept to be thrown in by the user, the PositionUpdateEvent object is now inserted into the event list for the first time.

CONCLUSION AND OUTLOOK

We have described the main points of the design and realisation of VisualRTJSim, a simulation framework that takes real-time Java programs, executes them in simulated time and enables observations and experiments. There are still small things to add or improve, but, more importantly, there also are some major subjects for follow-up research:

Programming-language theory has the notion of "abstract interpretation" (Cousot P. and Cousot R. 1977), meaning a shift in semantics that gives something less informative than the standard result of the program. Conversely, our addition of consume-blocks may be seen as an attempt at an "augmented semantics" that accounts for the physical effects that accompany the production of the computational result on a certain platform. Such a broadened, multidimensional notion of semantics seems necessary in real-time programming. Time is one but certainly not the only example (but time is special because of its strong relation to control). Energy consumption e.g. is a side effect of computation that often constitutes an important problem for the engineering of small embedded systems. So we will introduce a corresponding consume-statement into our simulation framework.

Another example, and closer to the traditional perspective on software, is computer memory, units of which can be claimed and freed in the course of a computation. Our present framework offers only trivial implementations of the RTSJ memory classes and thus cannot be employed to test for memory-related errors. We will next give a reasonably useful implementation of this chapter of RTSJ. An especially interesting possibility then is to plug real-time garbage collectors into the simulator. Real-time garbage collection is a subject of much current research; one needs predictable time consumption but also a rate of freeing memory space that is sufficient for the on-going allocation of program objects. See e.g. (Bacon, Cheng, Prajan. 2003) on this subject.

In its present state, our simulator assumes a single-processor system. For reasons of localisation and fault tolerance, many contemporary big-scale real-time applications, e.g. in avionics, use several computers on a network. RTSJ does not provide for this kind of applications, but does not preclude it either. We plan to evolve our simulator for use with distributed real-time systems also - which is no small task. The necessary modifications and additions reach from the re-interpretation of concurrency issues to means of representing network protocols in various degrees of accuracy.

Acknowledgements: This work started with the diploma thesis (Rosenkranz 2004). Java-based general-purpose discrete-event simulation systems are no rarity; see (Lechler and Page 1999) for a nice example in the Simula-tradition. We know of no simulator that specifically addresses RTSJ programs, but of course there is more work on simulating real-time systems in general. The hold_cpu procedure of ARTISST, a C-based simulation library for real-time systems, (Decotigny and Puaut 2001), gave the idea for our consume-method.

REFERENCES

Bacon, D., Cheng, P, Rajan, VT. 2003. "A real-time garbage collector with low overhead and consistent utilization." *Proc. Principlesof Programming Languages, ACM,* New Orleans.

Bernat, G., Burns, A., Wellings, A. 2000. "Portable Worst-Case Execution Time Analysis Using Java Byte-Code". *Proc. 6th International EUROMCRO Confernce on Real-time Sytems,* Stockholm.

Bollellella, G., Brosgol, B., Dibble, D., Furr, S., Gosling,J., Hardin, D., Turnbull., M. 2001. "The Real-Time Specification for Java; Version 1.0". http://www.rtjorg/rtsj-V1.0.pdf.

Cousot P. Cousot R. 1977. "Abstract Interpretation", *4th Symp on Principles of Programming Languages, ACM, 238-252,* Los Angeles, California.

Decotigny, D., Puaut, I. 2001 "*An Extensible and Modular Simulation Tool for Real-Time Systems*", Report 1420, Irisa, Rennes, France, http://www.irisa.fr.

Dibble, P., Wellings, A.: 2004. "The Real-Time Specification for Java: Current Status and Future Work." *Proc. 7th. IEEE .Symp. Object Oriented Real-Time Distributed Computing, 71-77.*

Lechler, T., Page, B. 1999. "DESMO-J: An Object-Oriented Discrete Simulation Framework in Java". *Proc. 11th European Simulation Symposium*, SCS-Publications, Delft.

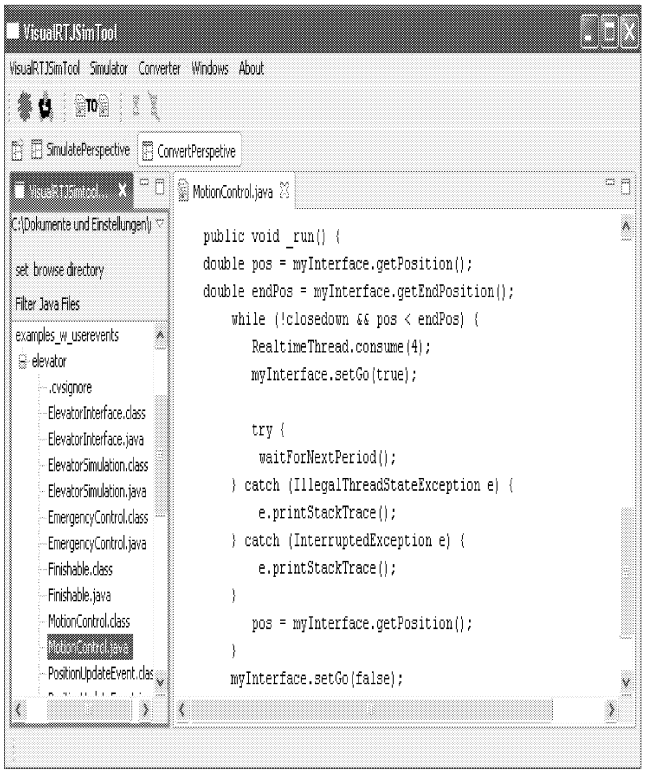
Rosenkranz, J. 2004. *„RTJSim - ein prozessorientierter Simulator für Echtzeit-Java".* Diplomarbeit, Naturwissenschaftliche Fakultät, Universität Salzburg.

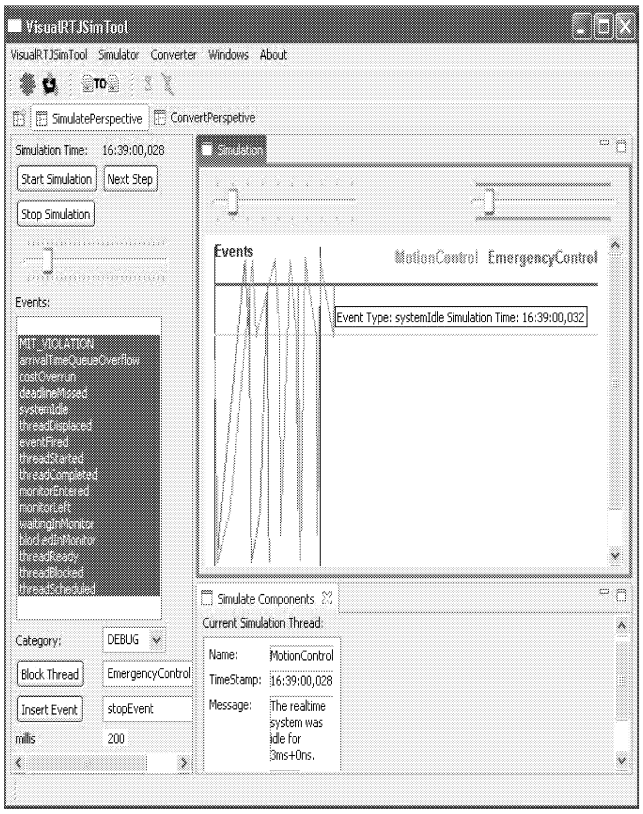
Rosenkranz, J., Pohlmann, W. 2005. „RTJSim: A Simulator for Real-Time Java". *Proc. ESM'05 - European Simulation Multiconference Porto,* (J.M.F.Teixeira, A.E.Carvalho-Brito ed.), EUROSIS-ET publ., Ghent, p. 48-50, Oct. .

Schoeberl,P. Pedersen,R. 2006. :"WCET Analysis for a Java Processor". *Proc. Workshop on Java Technologies for Real -Time and Embedded Systems* , Paris, Oct.. 2006.

Wellings, A. 2003. „Is Java Augmented with RTJS a Better Real-Time System.Implementation Technology than Ada95?" *Ada Letters,* 23(4), Dec. .

APPENDIX: SCREENSHOTS FROM VisualRTJSim





SIMULA

SIMULA TOOL FOR SIMULATION-BASED OPTIMIZATION

Jaroslav Sklenar
Department of Statistics and Operations Research
University of Malta
Msida MSD 2080, Malta
Web: http://staff.um.edu.mt/jskl1/
E-mail: jaroslav.sklenar@um.edu.mt

KEYWORDS
SIMULA, Optimization, Discrete Event Simulation.

ABSTRACT

The paper demonstrates the use of the SIMULA package OPTSIM that has been designed for teaching simulation-based optimization of discrete systems. Two optimization algorithms are implemented in several versions. For continuous optimization a modified Nelder-Mead moving simplex algorithm has been implemented. Local search algorithm is used for integer and mixed integer optimization. As OPTSIM is a subclass of another SIMULA class QUESIM for simulation of queuing systems, simulation based evaluation of objective and constraint functions is relatively easy. Two optimization examples are used to demonstrate the basic ideas.

INTRODUCTION

SIMULA (SIMple Universal LAnguage) is a general object-oriented language that together with two system classes SIMSET and SIMULATION introduced the principles of process-oriented discrete simulation. Being 40 years old, other object oriented languages are with few exceptions still conceptual subsets of SIMULA and most modern simulation tools except the special purpose ones basically implement the SIMULA process model. For basic facts and discussion on some myths about SIMULA see (Sklenar 2003) that also shows how modularity is achieved by the SIMULA's object-oriented model itself without using any other facilities like header files (C++), packages (Java) or units (Pascal). Reader interested in history is referred to the paper (Holmevik 1994).
SIMULA can be applied in a large spectrum of simulation related problems ranging from the most advanced simulation modeling tasks like for example Reflective Simulation of systems that contain their own simulation models to introductory courses introducing basic ideas of computer simulation. The author has been using SIMULA as the first simulation language in a course that introduces simulation modeling to IT students. For this purpose a package called QUESIM has been written. It supports simulation of queuing systems. QUESIM has been created in order to add some general and some special purpose tools while keeping all facilities of the original class SIMULATION directly available without any restrictions. So the user can use the standard SIMULA facilities for work with parallel processes that by the author's opinion represent the most advanced and the most natural way of creating discrete simulation models. For more details see (Sklenar 2005). In this paper we describe the tool called OPTSIM intended as a support for simulation-based optimization.

SIMULATION-BASED OPTIMIZATION

Any deterministic single objective optimization problem can be generally expressed as

$$\min\{f(\mathbf{x}) \mid \mathbf{g}(\mathbf{x}) \le \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$$

where f is the scalar multivariate objective function and vector multivariate functions $\mathbf{g}$ and $\mathbf{h}$ represent the inequality and the equality constraints. Maximization of f is in fact minimization of $-f$. There are very many optimization algorithms that have been developed to solve the above problem. For their overview and the background theory see for example the book (Bazaraa 1993) to mention one out of many. A common assumption of all optimization algorithms is that for a given solution vector $\mathbf{x}$ the functions f, $\mathbf{g}$ and $\mathbf{h}$ can be evaluated. This is true for all textbook examples, but unfortunately the reality is different. For many especially stochastic systems we know that a certain function exists, but no algorithm is available for the function evaluation, so the only option is simulation. This is the essence of simulation-based optimization that compared with classical optimization theory brings its specific problems and restrictions. For a more detailed discussion of this topic see for example (Azadivar 1999).

Feasibility

Mostly simulation doesn't provide, because of obvious reasons, exact results. Of course we can increase the number and duration of simulation experiments and apply some variance reduction technique, but this can only decrease the error, not eliminate it. This simple fact makes the use of equality constraints such that the functions h_i are evaluated by simulation very questionable. The same applies to tight inequality constraints g_i. The result is that we have to accept a certain tolerance on equality; otherwise the solution vectors would all be infeasible with respect to equality constraints and possibly infeasible for tight inequality constraints. This applies generally even to constraints whose functions can be evaluated without

simulation. The difference is the value of the tolerance that in case of known functions is under user's control as one of general tolerance related optimization parameters of the tool used (see for example Matlab Optimization toolbox).
Another problem related to feasibility is the fact that infeasibility may not be revealed before starting the simulation. So it may happen that we simulate for example unstable queuing systems where crash sooner or later inevitably happens. Thus it is very important to try to check feasibility before actually starting simulation. Of course it may be impossible. Then the simulation model should contain tests to detect infeasibility. For example we can define an upper bound on queue length and stop the simulation if it is reached. A general guide hardly exists; we have to make use of any knowledge available about the system simulated.
So far our assumption was that feasibility is tested by the values of the functions g_i and h_i. Alternatively, depending on the optimization algorithm used, it is possible that the simulation experiment reports infeasibility by a large objective function value (*maxreal* in SIMULA). We then solve an unconstrained problem for which only evaluation of the objective function f is needed, provided the optimization algorithm copes with possibly infinite objective values.

Algorithm selection

There are very many optimization algorithms and ways of their classification and assessment. A common criterion for classification of optimization algorithms is the type of information about the objective function that the algorithm is using. There are two classes of algorithms: those that use the derivatives of the functions and those that don't.
Algorithms based on derivatives are generally superior because more information is available about the function behavior. The obvious requirement is differentiability of the functions involved necessary to compute the gradients. Even for known functions there are problems with gradient based methods. Though intuitively attractive, the original gradient (steepest descent) algorithm is in fact one of the worst algorithms for multivariate optimization. There are two reasons for this fact. First the gradient moves are orthogonal. This results in zig-zag effect. The other reason is possibly a very small norm of the gradient, for some functions even far from the minimum, which makes the moves very short. The result is that "long flat valley" functions require thousands of univariate search steps before the minimum is reached. There are many improvements to the original gradient search algorithm. Orthogonality of minimization directions can be overcome by various conjugate gradient methods. Another group of methods called pseudo-Newton methods use gradients to approximate the Hessian matrix, so also some information about the second derivatives is utilized. In the context of simulation-based optimization the gradients are unknown. Great effort has been invested to find the approximate values of the gradient by simulation. Methods are based on a common idea: evaluate the function at perturbed points and use the differences to compute the gradient approximately. For more see for example (Spall 1998). All these methods suffer from the other problem. Close to the minimum the norm of the gradient can be very small, much smaller than simulation error. Thus the results may be completely misleading, for example we may obtain wrong minimization directions.
Fortunately there are optimization algorithms that don't need gradients, for overview see (Bazaraa 1993). One of the most popular methods from this group is the so-called Nelder-Mead moving simplex method introduced in (Nelder-Mead 1965). This method should not be mixed with older and better known simplex method for solving linear optimization problems. Detailed description of the method is beyond the scope of the paper, we show just the basic ideas by figure 1 (created by Matlab) depicting first 9 steps of minimization of a function with minimum at (2 1). Simplex is a polytope made of n+1 points in n dimensions, so a triangle for n=2. In the figure the starting simplex is [(0 1), (0 2), (1 2)]. The first operation is the so-called reflection from the worst point (0 2) over the average of the remaining points – here (0.5 1.5). With reflection coefficient 1 the tested point was (1 1). It was found good, so the move was extended by the so-called expansion to get the second (red) simplex. In the next step the expanded point was worse than the reflected one, so the result was a reflected (black) simplex. In the following steps the reflection was never good, so the algorithm performed the so-called contraction towards the middle points (green simplexes). The 9th step resulted in another reflection, etc. Operation not shown in the figure is shrinking the simplex towards the best point if all attempted moves fail.

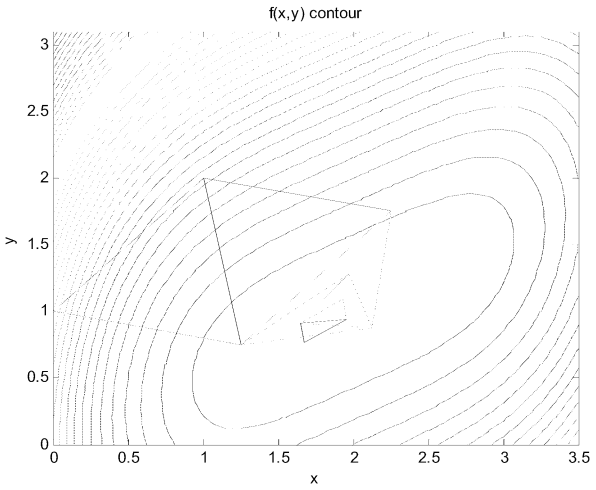


Figure 1: Moving simplex visualization

The algorithm is very robust; all we need is function evaluation at relatively not many points as compared with algorithms that include line search steps converging only asymptotically to their respective minima. So it seems to be very attractive in our simulation-based optimization context where number of function evaluations (simulations) should be minimized. On the other hand

there is very limited theoretical background for this algorithm. An exception is the paper (Lagarias - Reeds 1998) that gives convergence conditions, but for low dimensions only. Like all non-linear optimization algorithms, Nelder-Mead can get stuck in a local minimum. The standard approach to handle this is to restart the algorithm with a new simplex. There is another very important advantage of the algorithm in our context. It copes with "infinite" (*maxreal*) values of the objective function. Such points are considered as the worst ones. The algorithm then moves away that keeps feasibility of all simplex points. We assume that the initial simplex has mostly one infeasible point.

In OPTSIM a modified version of the algorithm has been implemented. First a second expansion is allowed to speed up the move together with the possibility of moving the whole simplex. For some functions a considerable improvement has been obtained, but more experimentation still has to be done. Of course all coefficients and modes of operation are under user's control.

OPTIMIZATION EXAMPLE

Let's consider a single queue single channel system with uniform distribution of inter-arrival intervals in [0,a] and uniform distribution of service duration in [0,b]. The values a and b are under our control. Arrival rate can be decreased, the service rate can be increased with associated costs $c_A a$ and c_B/b respectively. Then we consider waiting cost $c_W L_Q$ where L_Q is the average length of the queue. The constants are unit costs per appropriate time unit. We want to find such values of a and b that would minimize the total cost, so we solve the problem:

$$\min\{c_W L_Q(a,b) + c_A a + c_B / b \mid a > b, a > 0, b > 0\}$$

The first constraint guarantees stability (traffic rate $\rho<1$). Moreover we assume an unconstrained ("not at a boundary") minimum because otherwise there would be no solution due to the open feasible set. The function $L_Q(a,b)$ can only be evaluated by simulation because we have a G/G/1 system with continuous distributions. The following is the structure of the SIMULA program that finds locally optimal values of the parameters a and b:

```
Begin

 declare and read data (n = dimension)

begin
 external class optsim;
 optsim(n) begin

  real procedure myobjective(x,counter);
    real array x; integer counter;
  begin
   repeated simulation experiment
  end;

  boolean procedure feasible(x);
    real array x;
  begin
    feasible := x(1)>0 and x(2)>0
                and x(2)<x(1);
  end;

  initialize(insim,epsil,alpha,beta,gama,
             delta,mode);
  result := neldermead3(myobjective,
                        feasible,true);
display the results

end;end;end;
```

Data read from a text file is the problem dimension (here n=2), the initial simplex, values of all coefficients and mode flags. The problem dimension is used as a parameter of the class OPTSIM by which we prefix a block. This imports all simulation and optimization facilities including creation of various arrays used by optimization whose size depends on the problem dimension. After reading the data, the next statement calls the OPTSIM procedure *initialize* that prepares the optimization. The parameters are the initial simplex, the search termination precision (norm of the difference between the best and the worst simplex points), the reflection, contraction and expansion coefficients, ratio for simplex move after second expansion and mode flag whether to use enhanced search mode. The next statement calls the actual search. The first parameter is a real procedure that returns the objective value. Note that the Nelder-Mead algorithm procedure passes two arguments: the solution vector **x** and the counter with the current number of objective evaluations. This may be used for example in status reporting that is almost necessary for long computations. The objective function real procedure (here *myobjective*) returns the objective value, so OPTSIM can be used also for solving optimization problems that don't need simulation. Feasibility of a selected solution **x** is first tested by a boolean procedure, here *feasible*, that is passed to the optimization procedure as the second argument. We note that the first element of the solution vector is the maximum interval a, the second one the maximum service b. The procedure would return true for unconstrained problems. The third optimization argument is boolean, if it is true the procedure *neldermead3* will create a log file with all simplex points and their objective values. The following is an extract from the log file for c_W=1, c_A=10, c_B=100.

```
Evaluations:     3
Simplex: 2.00      3.00      2.50
         1.00      1.00      2.00
       120.15    130.05     76.00

Evaluations:     4
Simplex: 2.00      2.625     2.50
         1.00      1.25      2.00
       120.15    106.38     76.00
```

```
Evaluations:    39

Simplex: 3.45      3.45      3.44
         3.10      3.11      3.10
        69.29     69.29     69.24
```

We see the initial simplex after 3 evaluations, the next (contracted) one and the result after 39 evaluations. Each evaluation was done by 10 repeated experiments with duration 60000 time units each. Optimization parameters were: precision 0.01, reflection coefficient α=1, contraction coefficient β=0.5, expansion coefficient γ=2, basic mode without second expansion and simplex move. Optimization was repeated for various starting simplexes with same results within the given precision, so the objective function might be convex. The repeated simulation itself within the procedure *myobjective* is standard. OPTSIM is a subclass of QUESIM, so creating simulation model of queueing systems is rather simple. The following is the almost complete code.

```
real procedure myobjective(x,counter);
        real array x; integer counter;
begin
ref(tally) lq; ! queue length statistics;
 for i:=1 step 1 until 10 do begin
 external class quesim;
 quesim begin          ! simulation block;
  ref(station) ggc;    ! service station;

  g_customer class customer;
  begin          ! life = getting service;
    ggc.service(current,uniform(0,x(2),
                seedserv));
  end of process customer;

  process class generator;
  begin
    while true do begin
      activate new customer(0,0);
      hold(uniform(0,x(1),seedarr));
    end while;
  end of process generator;

  ! Simulation block body:;
  ggc :- new station("G/G/1",
             new fifo("Q",maxint),
             new multichannel("S",1,0,0),
             new tally);
  activate new generator;
  hold(60000);
  lq.update(ggc.LQ);
 end for;
 myobjective := lq.average*waitcost +
                servcost/x(2) +
                arrivalcost*x(1);
end of myobjective;
```

Note that the tally *lq* is updated 10 times, its average is taken as the average queue length L_Q. Not shown is the feasibility test. If the queue length reaches a big value (1000), simulations are stopped and the procedure returns the *maxreal* value. So we have a feasibility check before simulation by the procedure *feasible* and then during the simulation. The first one saves computing time, the second one is important to avoid crashes for values of traffic rate close to 1. Generally there may be other feasibility requirements that can be tested only by simulation like for example not exceeding a given average waiting time, etc.

INTEGER OPTIMIZATION EXAMPLE

We consider a hypothetical queuing network shown in Figure 2. Arrival to the network is Poisson with the rate λ=1 customer per minute. The stations are multi-channel with mean service time 3, 7 and 5 minutes respectively. The actual distribution is triangular with minimum and maximum values mean -/+ 50% of the mean respectively. After service in the station 1, 75% of customers proceed to the station 2, the rest go straight to the station 3. We have 20 personnel available that should be allocated in such a way that would minimize the total average number of waiting customers. This system can be for example an abstraction of a bank where customers are served by receptionists, tellers and cashiers.

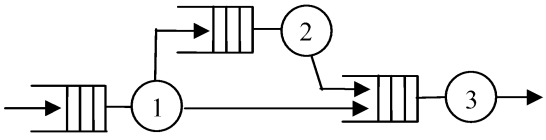


Figure 2: Example queuing network

So with all the parameters mentioned above fixed we solve the following constrained problem in three integer variables:

$$\min\{L_Q(x_1,x_2,x_3) \mid x_1 + x_2 + x_3 \le 20\}$$

where x_i is the number of channels in station i and L_Q is the total expected number of waiting customers. Due to the distribution of the service time the function L_Q can only be evaluated by simulation.
The structure of the SIMULA program that finds a locally optimal solution is almost identical to the one discussed above. The change in the procedure *feasible* is obvious, the actual activation of the optimization algorithm is this:

```
result := localsearch3(myobjective,
                       feasible,x0,false,true);
```

The local search procedure is given 5 arguments. The first one is the procedure that returns the objective value. It has two arguments, the integer solution vector and the counter with evaluation number. The second argument is a boolean procedure that checks the feasibility before objective evaluation. In this example it tests whether the sum of the vector elements is less or equal to 20. The third argument

of the procedure *localsearch3* is a feasible starting point. Next two boolean arguments turn on the debugging mode (stop after each tested point) and logging all tested points in a text file. The local search algorithm is standard, the tested neighborhood is the "cross":

$$N_1(\bar{\mathbf{x}}) = \{\mathbf{x} : \sum_{i=1}^{n} |\bar{x}_i - x_i| \leq 1\}$$

The search moves to the first better point in the neighborhood and stops if there is no better feasible neighborhood point. The following is an extract from a log file. The first three figures are the numbers of channels, the 4th one is the rounded objective obtained by 10 runs, each 60000 minutes long.

```
 Evaluation 1     4   8   6    2.113
 Evaluation 2     5   8   6    1.760
 Evaluation 3     6   8   6    1.658
 Evaluation 4     6   7   6    1.835
 Evaluation 5     6   8   5  122.023
f(at local optimum) = 1.658  x: 6   8   6
```

Another log file shows a well known fact that reaching a global optimum is not guaranteed:

```
 Evaluation 1     5   7   5  113.407
 Evaluation 2     6   7   5  118.605
 Evaluation 3     4   7   5  109.816
 Evaluation 4     3   7   5  142.775
 Evaluation 5     4   8   5  119.827
 Evaluation 6     4   6   5  118.618
 Evaluation 7     4   7   6    2.275
 Evaluation 8     5   7   6    1.919
 Evaluation 9     6   7   6    1.835
 Evaluation 10    7   7   6    1.800
 Evaluation 11    7   6   6    3.203
 Evaluation 12    7   7   5  118.303
f(at local optimum) = 1.800  x: 7   7   6
```

The simulation procedure is similar, the simulation block creates three stations, and the only important difference is the life of customers. Procedure *draw* returns true with probability 0.75.

```
g_customer class customer;
begin
 ggc1.service(current,
       triangular(1.5,3,4.5,seeds1));
 if draw(0.75,seedb1) then
    ggc2.service(current,
         triangular(3.5,7,10.5,seeds2));
 ggc3.service(current,
       triangular(2.5,5,7.5,seeds3));
end of customer;
```

Similarly to the Nelder-Mead algorithm, local search copes with infeasibility reported by a large value (*maxreal*). So it is possible to deal with both types of feasibility checks - before and during simulation.

CONCLUSION

Advanced and comfortable tools for visual interactive simulation have typically limited support for simulation-based optimization, if any at all. There are several reasons. First a collection of efficient optimization algorithms has to be implemented. Second the simulation models must be flexible enough to evaluate the objective value, but especially to detect infeasibility of input arguments during simulation and possibly to stop the simulation. The third requirement is flexibility in the interaction between the optimizing algorithm and the model including feasibility checks before starting simulation. Due to these reasons it is probably more convenient to opt for a programmed solution, especially when using a tool that makes programming of simulation models relatively easy and straightforward. SIMULA is an example of such a tool. It combines flexibility of a general high-level language with simulation programming comfort. The tool OPTSIM described in this paper achieves comfortable simulation by using the facilities of the simulation package QUESIM. For the optimization part several procedures for both continuous and integer, constrained and unconstrained optimization are available. Future development will be oriented to multicriterial optimization and heuristic algorithms.

REFERENCES

Azadivar, F. 1999 "Simulation Optimization Methodologies". In *Proceedings of the 1999 Winter Simulation Conference.* 93-100.

Bazaraa, M.S.; H.D. Sherali; and C.M. Shetty. 1993 *Nonlinnear Programming - Theory and Applications*, Wiley.

Gosavi, A. 2003. *SIMULATION–BASED OPTIMIZATION Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic Piblishers..

Holmevik, J.R. 1994. "Compiling SIMULA: a historical study of technological genesis". In *IEEE Annals of the History of Computing,* Vol 16, No 4, 25-37.

Lagarias, J.C.; J.A. Reeds; M.G. Wright; and P.E. Wright. 1998. "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions". In *SIAM J. OPTIM.* Vol. 9, No. 1, pp. 112-147.

Nelder, J.A. and Mead, R. 1965 "A Simplex Method for Function Minimization." In *COMPUT. J.* Vol. 7, 308-313.

Sklenar, J. 2003. "Object-Oriented Modularity of SIMULA". In *Proceedings of the 2003 European Simulation and Modelling Conference ESM2003*. 117-121..

Sklenar, J. 2005. "Simulation of Queuing Systems in QUESIM". In *Proceedings of the 2005 European Simulation and Modelling Conference ESM2005.* 35-37.

Spall, J.C. 1998 "An Overview of the Simultaneous Perturbation Method for Efficient Optimization". In John Hopkins APL Technical Digest, Vol 19, No 4, 482-492.

NEW FLEXIBLE SIMULATION TOOL IN SIMULA

Ivan Krivy
Eugene Kindler
Department of Informatics and Computers
University of Ostrava
30. dubna 22, 701 03 Ostrava 1
Czech Republic
E-mail: ivan.krivy@osu.cz, ekindler@centrum.cz

KEYWORDS

Simulation, Object/oriented programming, non-standard simulation studies, SIMULA.

ABSTRACT

The modern simulation applications, namely those to intelligent system with sophisticated control techniques based on complex computer models, need communication among simulation models existing contemporarily. SIMULA offers a standard class *SIMULATION* for simulation but nowadays it makes problems, because it does not allow giving names to simulation models. Therefore, we decided to construct another class, similar as much as possible to *SIMULATION* but allowing identifying models. The new class is called *SIMULAT*. The paper contains its description and a discussion on the reasons for differences between both classes.

INTRODUCTION – NESTING MODELS

A habitual simulation study is organized as a cycle of simulation experiments. Thus an experiment begins only after the preceding one ends and no parallel existence of (mutually interacting) experiments are supposed. As a simulation experiment is an implementation of a simulation model, one can say that the common technique of simulation does not suppose the existence of more simulation models during a simulation study.

A different situation is met, when the simulated system *S* is an intelligent one, using e.g. an element *C* that generates decisions on the next operating of *S* by means of sophisticated computing, including simulation. *C* could test the logically possible decisions by simulation of their consequences in *S* and then use the simulated data for choosing the best decision. If *S* is simulated, its model *M* should contain an image of *C*, which creates "its private" models m_1, m_2,… that are similar but not equal to *M*. The similarity implies that all the mentioned models should be described by the same language (in the object-oriented programming implemented by the same classes). When a model m_i is constructed and initialized, a great amount of its components should be "copied" from *M* and, therefore, it would be suitable to distinguish m_i from *M*, since otherwise the languages for the description of both *M* and m_i should be the same. For example, how simple would be applying a statement like *m1.prod3.temperature:=M.prod3.temperature* to express that the instantaneous value of the *temperature* of the product *prod3* existing in *M* should be interpreted as the initial value of the *temperature* of the image of *prod3* in *m1*!

Unfortunately, the use of class *SIMULATION* does not allow giving names like *M* or *m1* to simulation models and therefore a simple way of copying values from one model to another is not possible and neither the other operations with them (like comparing). Special techniques were developed (Kindler 1999; Kindler 2000) for that purpose but they are rather sophisticated and an ordinary user could make errors using them.

Class *SIMULATION* is very suitable but it is based on the idea that a simulation model is a quasi-parallel system (Dahl et al. 1968; SIMULA a.s. 1989). The quasi-parallel systems allow applying statements like *resume(X)* that switch performing statements to the "life rules" of component *X* so that the possibility of return is neglected. If in a model *M1* statement *resume(M2.X)* was performed so that *M2* would be a model different from *M1*, after some steps the computing would enter a "wild run" and then collapse because of a nonsense instruction. In order to eliminate tests like that whether the current application of a *resume* statement does not lead out from the operating quasi-parallel system, SIMULA prescribes using the *resume* statements so that their parameters point to instances of classes local in true subblocks, and does not allow assigning names to subblocks. Therefore, the models based on class *SIMULATION* must be programmed as subblocks and cannot get names.

BASIC IDEA OF A NEW SIMULATION CLASS

Because of the mentioned difficulties, an idea arose to declare a class that would not use *resume* statement but behave like class *SIMULATION*. Using such a class would not suffer of the restriction concerning the model names.

In class *SIMULATION*, repeating use of *resume* statements is hidden, causing a sequence of "events" performed by stepwise resumed objects P_1, P_2, P_3, ... , where always P_i makes *resume* to P_{i+1}. In other words, it is always only P_i that "knows" what object will be P_{i+1} (it computes this "knowledge" during its life rules). Metaphorically said,

quasi-parallel system is a "jungle" of stepwise computing objects that are not liable to any "director", who would control their active steps. And thus the first idea is to replace the "jungle-like" communication *resume(P)* by something that would remind a feebling director, who works formally as a mighty organizer but who decides exactly according to the instructions sent him from his subordinated persons.

SIMULA offers statement *call(P)*. If it is performed by an object *R*, the behavior is similar to that of *resume(P)*, but – contrary to *resume – P* carries an information that he has been "called" by *R*. Performing its life rules, *P* may meet a statement *detach* that causes the computing process to return to the life rules of *R*. Contrary to *resume*, *call* statement does not need any special checks at which subblock, quasi-parallel system, etc. its parameter can be, and therefore, the path to such a parameter is not limited. Metaphorically said, an element *A* of a model can use *call* statement for demanding an element *B* of another model to serve *A* by performing a certain phase of life.

The principle seems to be clear and simple. P_i does not need to perform *resume*(P_{i+1}), instead it sends a message to the "director" in sense "I am ready, arrange P_{i+1} to continue". Moreover, SIMULA allows declaring a new procedure called *resume* in that sense. Namely, let the "director" be called *manipulator* and let a reference value *called* be for disposal; then procedure *resume(X)* could be declared as having body

```
begin called:-X; detach end;
```

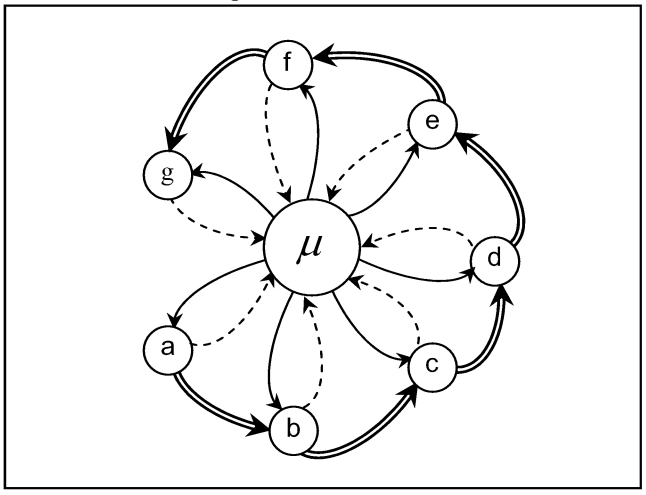


Figure 1: Role of manipulator

while the life rules for the *manipulator* would be a cycle

```
while true do call(called);
```

In Fig. 1 one can see an illustration how *manipulator* alternates with the objects. It is represented by the central circle while the user-defined objects are outlined by smaller circles *a*, …, *g*. Full line arrows represent *call*, dashed line arrows represent *detach* and arrows in double lines represent the resulting *resume*.

When *resume* is prepared as a "user defined" procedure in the outlined way, the declarations of the tools nested in class *SIMULATION*, presented in (SIMULA a.s. 1989), can be almost without change copied and nested into the new class *SIMULAT* that should replace class *SIMULATION*. Nevertheless, some obstacles remain; they will be presented in the next two sections, together with the technique how to overcome them.

ANALOGY TO ACTIVATION STATEMENT

The activation statement of SIMULA cannot be used outside subclasses of class *SIMULATION* and blocks prefixed by this class. Using the class that should replace *SIMULATION* excludes possibility to use the activation statement. Also the special keywords bound to the activation statement (*activate*, *reactivate*, *delay*, *at*, *before*, *after* and *prior*) are prohibited outside the context of the activation statement. Therefore, for every configuration of the activation statement, class *SIMULAT* should have a procedure that makes the same as the configuration should do; naturally, the name of such a procedure should recall the semantics of the configuration.

There were two reasons for introducing the activation statement into SIMULA in 1967. Firstly, the former users of the old simulation language used it because it was in that language (SIMULA I) and for them the retention of this statement was a good help to feel home when using the new SIMULA. Secondly, the number of "parameters" of activation statement varies and depends on its configuration and therefore, could not be replaced by a procedure, as the number of parameters of a procedure is uniquely determined by the declaration of this procedure.

Therefore, more than one procedure must be in class *SIMULAT*, each of which corresponding to a certain configuration of the activation statement. The base of the procedure title is *run* (instead of keyword *activate*) or *rerun* (instead of keyword *reactivate*). The base may be completed by a complement corresponding to a keyword used in the activation statement. The first reference expression of the activation statement becomes that performing the procedure while the other "parameters" become parameters of the given procedure. Table 1 informs how the configurations of the activation statements are replaced by the procedures.

Activation statement	**Corresponding procedure**
activate X	*X.run*
activate X at T	*X.run_at(T)*
activate X delay T	*X.run_with_delay(T)*
activate X at T prior	*X. run_priorly_at(T)* etc.
activate X after Y	*Xrun_after(Y)*
activate X before Y	*X.run_before(Y)*
reactivate X	*X.rerun*
reactivate X at T	*X.rerun_at(T)* etc.

Table 1: Activation procedures

All these procedures are declared with the use of procedure *ACTIVAT* (SIMULA a.s. 1989, section 12.2), which is, naturally, copied into class *SIMULAT* as well. The declarations are in Fig. 2.

MAIN PROGRAM AS A PROCESS

There is another problem that has not been solved by the tools described in the preceding sections. It concerns the fact that the "main program" of any subclass of class *SIMULAT* should behave as a process, namely respect the scheduling statements. This property of the main program is very useful in simulation practice, which is testified by analogies existing in many simulation programming tools.

```
procedure run;
  ACTIVAT(false,this process,"direct",0,none,true);
procedure rerun;
  ACTIVAT(true,this process,"direct",0,none,true);
procedure run_at(T); long real T;
  ACTIVAT(false,this process,"at",T,none,false);
procedure rerun_at(T); long real T;
  ACTIVAT(true,this process,"at",T,none,false);
procedure run_with_delay(T); long real T;
  ACTIVAT(false,this process,"delay",T,none,false);
procedure rerun_with_delay(T); long real T;
  ACTIVAT(true,this process,"delay",T,none,false);
procedure run_before(Q); ref(process)Q;
  ACTIVAT(false,this process,"before",0,Q,false);
procedure rerun_before(Q); ref(process)Q;
  ACTIVAT(true,this process,"before",0,Q,false);
procedure run_after(Q); ref(process)Q;
  ACTIVAT(false,this process,"after",0,Q,false);
procedure rerun_after(Q); ref(process)Q;
  ACTIVAT(true,this process,"after",0,Q,false);
procedure run_priorly_at(T); long real T;
  ACTIVAT(false,this process,"at",T,none,true);
procedure rerun_priorly_at(T); long real T;
  ACTIVAT(true,this process,"at",T,none,true);
procedure run_priorly_with_delay(T); long real T;
  ACTIVAT(false,this process,"delay",T,none,true);
procedure rerun_priorly_with_delay(T); long real T;
  ACTIVAT(true,this process,"delay",T,none,true);
```

Figure 2: Declaration of activation procedures

At first sight, transforming the behavior of the main program to that of a process seems resisting to the main principles of SIMULA, as it projects the behavior of the main class to that of a class nested in it. Nevertheless, main class *SIMULATION* proves that it is possible and the way used in it can be applied in class *SIMULAT* as well.

This way consists in introducing a fictitious process called *main*, which owns its own items for using the scheduling statements, namely its event notice (SIMULA a.s. 1989, chapter 12). Its life rules are formed by a simple cycle *while true do detach*, which causes that any activation of that process switches the computing from *main* to the main program.

There are several consequences of this matter. Firstly, similarly as in class *SIMULATION*, *main* is the only instance of a hidden and protected class *main_program*, the declaration of which should be taken from *SIMULATION* to *SIMULAT* over. Secondly, the declaration of procedure *resume* should be extended just for the case of argument *main*:

```
procedure resume(p); ref(process)p;
begin detached:-called; called:-p;
    if detached=/=called then
    begin if detached=/=main
        then detached.detach
        else call(the_manipulator)
    end
end;
```

Thirdly, a reference variable *detached* should be introduced, as the procedure *resume* mentioned above runs with the only variable *called* (*detached* is intended to point to the object *P* that is just interrupting its "life", while *called* is intended to point to the object that should continue computing after *detached*). And fourthly, the life rules of class manipulator should be extended to the following form:

```
begin detach;
    while true do if called=/=main
    then call(called) else detach;
    end;
```

That causes that a "normal" process is called while *main* causes to switch to the main program (i.e. to the "life" of the given subclass of *SIMULAT*).

CHANCES OF CLASS *SIMULAT*

Class *SIMULAT* and its subclasses can be applied as block prefixes but that has no special effect – SIMULA system class *SIMULATION* can be used instead. Contrary to *SIMULATION*, it is possible (i) to generate instances of

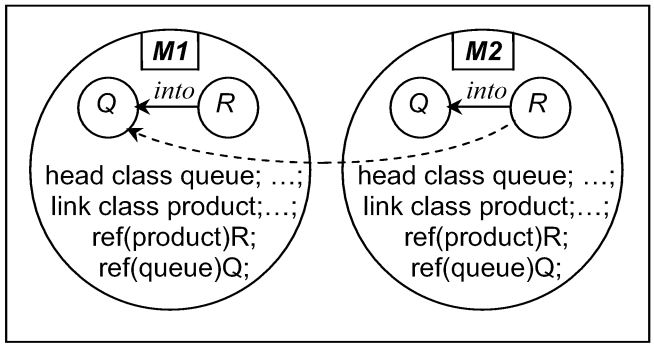


Figure 3: Illustration of transplantation error

any subclass *C* of class *SIMULAT* by applying object generator (see [4], paragraph 3.8) like *new C* and (ii) to assign the instances names (like *M:-new C*). The names enable two different models like *M1* and *M2* interact, although that possibility cannot lead to transplantation. If *R* is introduced in *M1* and *M2* as a pointer qualified to class *D* declared as nested in *M1* and *M2* as well, and *X* is e.g. an arithmetic attribute of *D*, then *M1.R.X:=M2.R.X* is allowed and useful, while the transplantation *M1.R:-M2.R* is refused, as *M1.R* is qualified into a class *D* local in *M1* and *M2.R* is qualified into a class *D* local in *M2*, i.e. into class belonging to a different prefix sequence. (See Fig. 3, where a more popular transplantation error is illustrated by ranging an element *R* of a model *M2* into queue *Q* of model *M1* – a statement like *M2.R.into(M1.Q)* – outlined by means of a dashed arrow – is discovered as an error as the argument of *into* should be an instance of class *Q* declared

inside *M2*.)

When a simulation model based on class *SIMULATION* concludes its operation, it disappears, because it must have been implemented as a block. A very stimulating property of class *SIMULAT* is that in a similar situation the block exhausting its performance can be present in the computing process (e.g. in a given simulation study) and later can be demanded to perform some procedures.

For that possibility, an idea offers to use some procedure that replaces the activation statement (e.g. *M1.run(main)* or even *inspect M1 do run(new C)* where *C* is local in *M1*). Supposing the *SQS* was not liquidated, then calling such a "run-like" procedure could "restart" a model that concluded its operation. In case *SQS* exists but is empty, the restarted model could begin up from zero model time.

That stimulates a further idea. Many discrete simulation models concern systems with a constant skeleton of permanent elements called base of model (Kindler 1978). In the simulation model, the image *J* of the base of the simulated system has often to be anew generated for every simulation experiment, as the classes of the components of *J* are linked by reference attributes with the processes and – if class *SIMULATION* is used – *J* disappears whenever the given simulation block is left. The use of class *SIMULAT* allows concluding a simulation experiment, passivating *main* and all elements of *J,* liquidating or passivating the other elements, i.e. empting *SQS* and leaving the model (e.g. by letting it make *detach* statement) but conserving *SQS* and the identifier of the model. Later a new simulation experiment can be performed with the same model (e.g. by using a "run-like" procedure), but its image *J* of the basis of the simulated system will remain from the previous experiment.

That technique favors the idea to modify procedure *passivate*: when *SQS* is emptied during its operation, let no run time error is identified: it would be better to replace it by statement *this SIMULAT.detach*.

```
SIMSET class simulat;
hidden protected SQS,EVENT_NOTICE,FIRSTEV,MAIN_PROGRAM,ACTIVAT,MANIPULATOR,detached, called,
  THE_MANIPULATOR,RESUME;
begin
  ref(head)SQS; ref(main_PROGRAM)main; ref(process)called,detached; ref(MANIPULATOR)THE_MANIPULATOR;
  link class EVENT_NOTICE(EVTIME,PROC); <copied from SIMULATION>;
  ref(EVENT_NOTICE)procedure FIRSTEV; FIRSTEV:-SQS.first;
  ref(process)procedure current; current:-FIRSTEV.PROC;
  long real procedure time; time:=FIRSTEV.EVTIME;
  link class process; hidden protected TERMINATED_; virtual: procedure initiate is procedure initiate;;
  begin ref(EVENT_NOTICE)EVENT; Boolean TERMINATED_;
    Boolean procedure idle; idle:=EVENT==none;
    Boolean procedure terminated; terminated:=TERMINATED_;
    long real procedure evtime; if idle then error("...") else evtime:=EVENT.EVTIME;
    ref(process)procedure nextev; nextev:-if idle or else EVENT.suc==none then none else EVENT.suc.PROC;
    procedure initiate;;
    initiate; detach; inner; TERMINATED_:=true; passivate; error("terminated called")
  end of process;
  procedure ACTIVAT(REAC,X,CODE,T,Y,PRIO); <copied from SIMULATION>;
  <procedure declarations presented in Fig. 2>
  procedure passivate;
  begin inspect current do begin EVENT.proc:-none; EVENT.out; EVENT:-none end;
      if SQS.empty then this SIMULAT.detach else resume(current)
  end;
  procedure hold(T); <copied from SIMULATION>;
  procedure wait(S); <copied from SIMULATION>;
  procedure cancel(X); <copied from SIMULATION>;
  process class MAIN_PROGRAM; <copied from SIMULATION>;
  procedure accum(a,b,c,d); <copied from SIMULATION>;
  class MANIPULATOR; begin detach;  while true do if called=/=main then call(called) else detach end;
  procedure RESUME(p); ref(process)p;
   begin detached:-called; called:-p; if detached=/=called then
       begin if detached=/=main then detached.detach else call(the_manipulator) end
   end;
  THE_MANIPULATOR:-new MANIPULATOR; SQS:-new head; called:-main:-new MAIN_PROGRAM;
  main.EVENT:-new EVENT_NOTICE(0,main); main.EVENT.into(SQS);
end of SIMULAT;
```

Figure 4: Declaration of class SIMULAT

IMMEDIATE INITIALIZATION

SIMULA standard class *SIMULATION* introduced processes so that they are immediately detached. The consequence is that any process can initiatively operate only after it is activated and before its first activation a process is like a passive object able to operate only when getting a message to perform a procedure.

When constructing a simulation model, one often needs to generate a structure of processes (e.g. the machines in a production system) so that each of them should be initialized immediately after its creation and only when the whole structure is completed the processes should be activated. The initializing can consist e.g. of equipping the generated process by some evidence or inserting the process into a certain list. For that purpose, it would be suitable to declare a class of processes so that the use of a generator like *new C* (where *C* is the name of the class) would initialize the generated instance. *SIMULATION* appears thumby for satisfying such a demand. This situation was a challenge for class *SIMULAT*. Class *process* has a virtual procedure *initiate*, pre-declared as empty, that is performed before the first *detach* of the generated instance, i.e. immediately after its creation.

CONCLUSION

In 1997 Eugene Kindler, one of the authors of this paper, spoke with Ole-Johan Dahl about SIMULA and Ole-Johan made a short remark that he had another idea on SIMULA sequencing tools. Then we spoke on other aspects and Kindler never learned on the contents of the Ole-Johan's idea. May be it was exactly the same as one has discovered nowadays, ten years later.

Nowadays, the authors of the present paper map and test the consequences of the possibilities that class *SIMULAT* offers. It is mainly the security that has to be included into the class (observe for example what could happen if one applied statement *M1.hold(3)* in a situation that *M1* already finished its operation). The present state of class *SIMULAT* is outlined in Fig. 4. The full version of this class (according to the state described in the present paper) can be sent electronically to every interested person.

REFERENCES

Kindler, E. 1999. "Chance for Simula". In *Proceedings of the 25th Conference of the ASU – System Modelling Using Object-Oriented Simulation and Analysis*. ASU, Kisten, Sweden, 29-53.

Kindler, E. 2000. "Chance for SIMULA". *ASU Newsletter* **26**, No. 1 (May), 2-26.

Dahl, O.-J., B. Myhrhaug, and K. Nygaard. 1968. "Common Base Language". 1st edition. Norsk Regnesentralen, Oslo.

Simula Standard. 1989. SIMULA a.s., Oslo.

Kindler, E. 1978. "Classification of simulation programming languages, Part I – Declaration of necessary system conceptions". *Elektronische Informationsverarbeitung und Kybernetik* **14**, No. 10, 529-536.

AUTHORS BIOGRAPHY

IVAN KRIVY worked 13 years at the Nuclear Research Institute (1962-1975) and later (1975-1977) he worked at the Computer Art Establishment as a system programmer. In 1974 he graduated at the Charles University and in 1975 he obtained the degree Ph.D. in solid state physics. Since 1977 he was teaching courses in programming, probability theory and statistics at the Pedagogical Faculty at Ostrava. After the University of Ostrava having been established in 1991, he entered the Mathematics Department of the Faculty of Science and nowadays he works in the Computer Science Department as Professor in applied mathematics. Since 1992 his research activities are oriented to dynamic modelling and computer simulation (e. g. local epidemics, population dynamics) as well as to stochastic algorithms and their use in the global optimization. Krivy is a member of the American Mathematical Society, the Society for Industrial and Applied Mathematics, the Association of SIMULA Users, and the International Association for Statistical Computing. He is author of about 190 publications including 4 monographs. Stays abroad: Silesian University of Katowice (1991), Portsmouth Polytechnic (1991), University of Wroclaw (1995), Universidad de Cordoba (1996, 2000), and Lulea University of Technology (1999) and Blaise Pascal University of Clermont-Ferrand (2001, 2002).

EUGENE KINDLER was born in 1935 in Prague (Czechoslovak Republic). He studied mathematics at Charles University in Prague and there he got grades of Doctor of philosophy in Logic and Doctor of sciences in theory of programming. The Czechoslovak academy of sciences granted him the grade of Candidate of sciences in physics/mathematics. During his employment in the Prague Research Institute of Mathematical Machines (1958-1966), he participated at the design of the first Czechoslovak electronic computer and designed and implemented the first Czechoslovak ALGOL compiler for it. Then, working at the Ïnstitute of Biophysics at the Faculty of General Medicine of Charles University (1967-1973), he designed and imple-mented the first Czechoslovak simulation language and then introduced the object-oriented programming into Czechoslovakia. Nowadays, as professor emeritus of applied mathematics, he teams up with University in Ostrava and with Charles University in Prague. He was visiting professor at the University in Italian Pisa, at the University of South Brittany in French Lorient, at Blaise Pascal University in French Clermont-Ferrand, and at West Virginia University in American Morgantown. His main interest consists in object-oriented and modeling of systems that handle complex models to improve their own antici-patory abilities and intelligence. He applied it namely in logistic and production systems, for example in partici-pating at two grants supported by the European Commission and oriented to modernizing sea harbors by means of computing technique. Beside computer science, he is interested in music. He plays violin, piano and organ and for 30 years he is a director of a singing group performing the Latina, Greek, Armenian and Palaeoslavic chants of the first millenium.

SIMULA FEATURES, EXTENSIONS AND SIMULATION TOOLS

Alain TANGUY
LIMOS, CNRS UMR 6158, Université Blaise Pascal,
Complexe des Cézeaux, 63173 Aubière Cedex, France
E-mail: tanguy@isima.fr

KEYWORDS
SIMULA, Client/Server Architecture, Graphical User Interface, Simulation, Queueing Network

ABSTRACT

This paper is an attempt to answer the question: How to do modern programming using the SIMULA language? It contains a list of available compilers for popular operating systems, their particularities or limitations and a description of the main SIMULA features. Then some modern extensions are described, among them: graphical user interface, client/server architecture, database and callback. The 40 years of SIMULA programming provide a lot of simulation tools based on SIMULA. Finally, modelling experiments are presented and discussed.

INTRODUCTION

The first object-oriented language SIMULA was presented in 1967 at the IFIP Working Conference on Simulation Programming Languages. It is the fruit of previous works concerning parallel processes and simulation (Dahl and Nygaard 1965, 1966). The coroutine and block orientation is based on the ALGOL language (Backus et al. 1960).

The name SIMULA may be understood as SIMple Universal LAnguage as well as SIMUlation LAnguage. Three names are associated to its design: O.J. Dahl, B. Myhrhaug and K. Nygaard. A lot of books and works have been published. So, valuable documents, contributions and useful links can be found at Jean Vaucher's Web site *www.iro.umontreal.ca/~ simula/* and this of Jaroslav Sklenar *staff.um.edu.mt/jskl1/asu/*. The site of the ASU (Association of SIMULA Users) is located at *www.isima.fr/asu/*.

This paper presents main features of the SIMULA language, essentially the notions of block, class, coroutine and simulation process. An object is always dynamically created and begins its life as defined by its body coroutine. Then the modern SIMULA compilers are listed. The second section describes many extensions by means of C interfaces and external classes. Lastly, simulation tools, simulation models and experiments are presented.

SIMULA LANGUAGE

The SIMULA language (SSG 1989) is object-oriented and block-oriented. A *block* may contains local declarations, local instances, subblocks and statements. An object is an instance of a *class* composed of:

- an *attribute* part, that contains quantities, object references and methods: procedures or functions (normal or virtual);
- a class *body* (or *life rules*), that acts like a block describing the object behaviour and it is executed as a coroutine (Knuth 1969) when a new object is created.

The class nesting is implemented by inner classes as in the Java language. Excepting the basic simple types boolean, character, integer and real the others are implemented as dynamically generated objects. The character strings are implemented as *Text* objects. The created objects are manipulated by means of references, standing for a kind of pointer. For safety reasons, the created objects are freed by a garbage collector when it is necessary.

Basic Features

A SIMULA program can be seen as a hierarchical collection of blocks (SSG 1989). The hierarchy is induced by declaration nesting and inheritance relation. A block instance is said to be local to the one which directly contains its describing text. This relation defines a textual enclosure. The classic single inheritance relation is completed by split body nesting. The equivalent effective body of an object is built including recursively body in place of *inner* statement in the body of the direct ancestor.

The constituent parts of a program execution are dynamic instances of blocks; subblocks, prefixed blocks, connection blocks and class bodies. A block may be prefixed by a class. A connection block is created by the *inspect* connection statement, in order to give implicit definition to some items. The SIMULA language provides two statement sets for block instance sequencing as coroutines, one for semi-symmetric sequencing (*new*, *detach*, *call*) and an other for symmetric sequencing (*detach*, *resume*). The *new* statement generates an object and starts the execution of the object body. The object execution is suspended by the *detach* statement and may continue when another object executes *call* or *resume* on it. A block instance is at any time in one of the four states of execution: attached, detached, resumed or terminated, as shown in figure 1. A non-class block is always in the attached state.

The declaration nesting induces the *Quasi-Parallel System* notion (*QPS*). A *QPS* is identified by any instance of a subblock or a prefixed block, containing a class declaration. Such a block is called the *system head*. A *component* is a nested structure of block instances. The outermost block of a component, which identifies the component, is called the *component head*. The system head is the head of the *main*

component of the system. The heads of *object components* are those of detached or resumed objects which are local to the system head. An attached or terminated object has no component head status. The sequencing is based on management of the operating chain and reactivation chains. The operating chain is the sequence of blocks dynamically enclosing the currently executing block. A block of this sequence is attached to the previous one or it is a resumed object and the head of its system is attached to the previous block. When an object *X* executes a *detach* statement the part of the operating chain dynamically enclosed by *X* becomes the reactivation chain of *X*. The *call* and *resume* statements, operating on *X,* insert the reactivation chain of *X* into the operating chain. Block termination and detach statement involve implicit sequencing depending on block state and *QPS* nesting. The Mejtski's diagrams (Kindler 1995) help to specify the nesting of the objects and to understand their dynamic evolution.

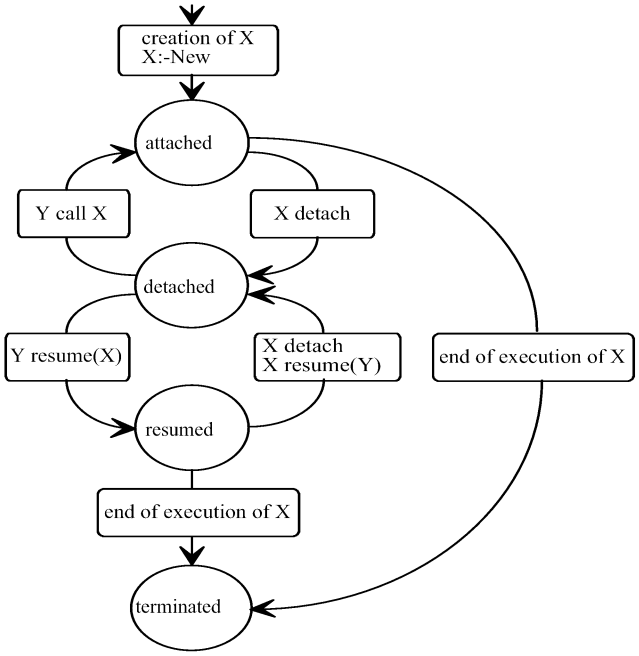


Figure 1: State Diagram of the Object X

Among the predefined classes, the *SimSet* class implements the bidirectional linked lists and the *Simulation* class provides discrete event simulation objects. As described by Dahl (Dahl 1968), the basic *process* objects are associated to event notices inserted into a sequencing set. The sequencing procedures (*activate*, *hold*, *passivate*, *wait*, *cancel*) synchronize the quasi-parallel operations of *process* objects. The *accum* procedure computes the integral of a step function of system time. Random drawing following a given probability distribution is defined by procedures of the system class.

The basic file class includes text and binary files using sequential and direct record access. Other extensions are introduced by external classes and libraries provided by the compilers and the SIMULA community.

Compilers

The first SIMULA standard 1967 was revised in 1986-1987 (SSG 1989). The Norwegian Simula a.s. commercial compilers were realized for Intel x86 DOS, IBM OS/2 and UNIX systems. A DOS demonstration version and a free one for IBM360/370 are available. Of course, the DOS versions work well under Windows. The Swedish Lund Software House AB commercial compilers were realized for UNIX systems and Mac. A demonstration version is available for Macintosh under Mac OS. A new version for Linux was developed in 2002. Andras Gaspar realized a Hungarian compiler for DOS, it is presently free. The Norwegian CIM compiler *folk.uio.no/cim/* is free too and open source. It is a preprocessor SIMULA to C, its target is UNIX or UNIX like systems that includes Linux, DOS with DJGPP extender *www.delorie.com/djgpp/*, Windows with MinGW *www.mingw.org/* or Cygwin *cygwin.com/*. These libraries allow a program to execute UNIX functions. The MinGW library is minimalist GNU for Windows, Cygwin is more sophisticated and enables to use Fluxbox or X server *www.fluxbox.org/* or *x.cygwin.com/*. P. Novak enhanced a CIM compiler for MinGW under Windows.

Almost all the compilers provide an interface with assembly and C language. Some automatic type conversions between SIMULA and C are also implemented, they mainly concern character strings and arrays.

Excepting Scandinavia, Canada and some European countries, SIMULA has never been a widely spread language. One may list some critical points (Sklenar 1997):

Generally
- Born in a small European country
- Old and frozen from 1968 to 1987
- Too complicated for the market needs of the 70's
- Expensive, no modern IDE
- Not enough publications and users

Language features
- Limited file access facilities (typed files)
- Missing data types (records, sets)
- No advanced parallelism, threads and real time support
- No GUI support

OOP features
- No multiple inheritance
- No interfaces

Simulation
- No automatic collection of statistics
- No report generator
- No useful specialized facilities (resources)

The five first points are surely at the origin of the slow spreading of SIMULA during the 70s and the 80s. We may remark that the new standard 87 and new compilers introduced modern IDE (for instance: Simula a.s. PC compiler). GNU projects and free software provided flexible development environments (Crimson editor, SciTe, KDevelop...) . The multiple inheritance is forbidden so we have to use aggregation and abilities. As we will see in the next sections, the language features and simulation lack problems were largely solved using an interface to the C or assembly language and the design of external classes acting

as language extensions; for instance: the *Demos* class (Birtwistle 1979). It is easy to launch operating system processes but it is more difficult to implement and manage threads and interrupts inside a SIMULA program. The real time problems are less important when we regard simulation and simulated time axes.

New languages appeared on the market: C in 1972, C++ in 1985, Java in 1995, graphical simulators and many less greatly known languages. When the OOP was needed and despite of its qualities SIMULA had difficulties to compete with software tools having material or operating system implementation bases (UNIX for C and C++, SUN for Java...). It was not American, a bit expensive, frozen for a long duration and not largely known.

SIMULA EXTENSIONS

SIMULA provides linked lists thanks to the class *SimSet*. In 1990, B. Kirkerud implemented data structures in the external class *SetTools*. It includes the classes *Container*, *Sequence*, *Set*, *Stack*, *Bag*, *Table* and *Tabled_Set*. In 1995, J. Vaucher designed the class Containers that implements the classes *Queue*, *Sequence*, *Priority_queue*, *Heap*, *Stack*, *Dictionary* and *Table*.

The external class *WindowTools*, designed by B. Kirkerud 1989 and revisited by J. Vaucher 1995, implements a sophisticated Graphical User Interface (GUI) for X servers. It provides multiple windows, buttons, text and edition box, graphic primitives... The automation of component alignment facilitates the design of windows, as shown in figure 2. It is also possible to program an animation of a simulated model. Furthermore, this class could be rewritten to manage a MS Windows graphical interface, for instance the wxWidgets library *www.wxwidgets.org/*. For 3D graphic animations a SIMULA to C interface to OpenGL or SDL may be foreseen.

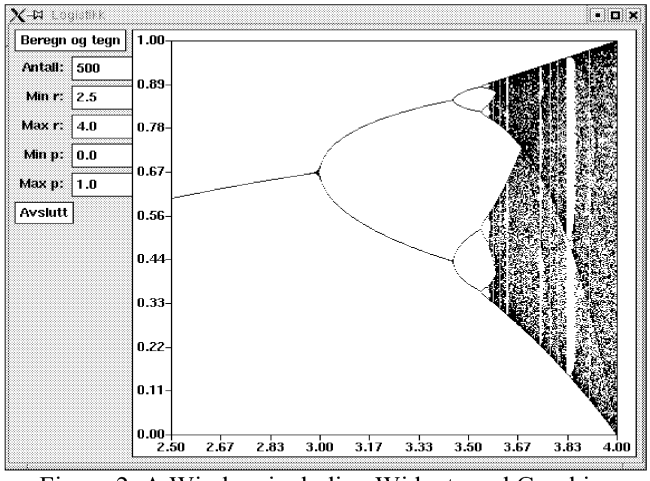
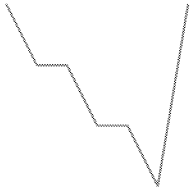


Figure 2: A Window including Widgets and Graphics

Presently, many applications generate sophisticated documents, drawing or pictures, for instance Doc, Xls or Pdf formatted files. Of course, we may prefer open formats such as Odt (Open document text or Odg for graphics), Xml or SVG (Scalable Vector Graphics). The following SVG file draws a small red stair defined by a *polyline* primitive and a sequence of points, the colour is given by the *style* and the graphic object is translated using a *transform* primitive. Figure 3 shows the result.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
  xmlns="http://www.w3.org/2000/svg">
<polyline points="0,0 10,20 20,20 30,40 40,40 50,60 60,0"
  style="fill:white;stroke:red;stroke-width:2"
  id="polyline4"
  transform="translate(36.28833,25.12272)" />
</svg>
```


Figure 3: Simple SVG Graphics

A lot of primitives are available so as to draw rectangle, ellipse and almost any kind of objects. A SVG file may be compressed to reduce its size. Inkscape is a GPL editor for SVG files. It is able to convert SVG files to different formats, for example to Odg that can be read by OpenOffice and many other text processors.

From 1992 to 1997, J. Vaucher enhanced the external class *Sockets* for UNIX systems. Its main application domain concerns networking, inter-process communication and client/server architecture. That enables a SIMULA program to perform a complex task using a specific server written in any suitable language. For instance, a simulation model, written in SIMULA, sends the new coordinates of a simulated object to an animation server written in Delphi. Of course, it is necessary to define a simple communication language and a protocol. Since 2000, we develop a *WinSockets* external class for SIMULA programs under Windows. It is not identical to the first one because the new Windows sockets are non blocking and MS is known not to respect POSIX standards.

In 1996, J. Vaucher implemented the class KeyFile based on access by key and variable record length. The Montreal classes may be tested using the CIM compiler under UNIX or Linux. The library of the CIM compiler must be corrected when it works under Windows with MinGW or CygWin. The size of *directfile* record must be one character longer because the end of line mark, under Windows, is Carriage Return + Line Feed instead of only LF for UNIX.

Some months ago, with K. Tanguy, we tested the CIM3.37 compiler under Linux Gentoo on Athlon64X2 dual core 4200+ so as to implement a very simple application in SIMULA that reads a database. The engine is SQLite3 *www.sqlite.org/*. The data table is composed of three fields:

Name (key, string), *Age* (integer) and *Weight* (real). So we handle basic data types. The SIMULA program calls the C function *dbr* that sends the SQL queries in order to read the data. SQLite returns each record in calling a C callback that stores the ith record into the ith row of three arrays. These arrays are global variables of the C module. When all the records are transferred, the function *dbr* continues its execution, closes the database and returns the control to the SIMULA program with the number of records. The C function *getrec* returns the i^{th} record from the global variables of the C module. It is used inside a loop of the SIMULA program to display the collected data. The following code defines the interface:

```
External C procedure dbr is
  integer procedure dbr(mydb, mycmd);
  text mydb, mycmd; ;
External C procedure getrec is
  integer procedure getrec(n, Nname, Nage, Nweight);
  Name Nage, Nweight;
  integer n;
  text Nname; integer Nage; real Nweight; ;
```

The parameters *mydb* and *mycmd* pass to the C function the database name and the SQL query.

These extensions demonstrate that the SIMULA language is not limited to simple operations for obsolete software. The next section provides information about the modelling and the simulation in SIMULA.

SIMULATION TOOLS

The classes *SimSet* and *Simulation* belong to the SIMULA language. The compiler must translate the *ACTIVATE* statements into *Activate* procedure calls. In practice, it is sufficient to implement process-oriented simulation models but the modeller must collect the statistics data and compute the performance criteria. Many external classes have been designed to implement different kinds of models.

External Classes

The class *Gpsss* was designed in 1971 at Montreal University. It implements GPSS like modelling, for process-oriented discrete event simulation, with all the abilities of an object-oriented language. It is a Swiss knife for queueing network models that offers main objects:

- *Transaction*, a customer;
- *Facility*, single server queue with priority;
- *Storage*, multiple server, stock or resource;
- *Region*, statistics area;
- *Group*, for treatment of fixed size set of customers;
- *Table* of data;
- *Wait_monitor* process for the management of transactions waiting on a particular condition specified by the *Wait_until* method.

A customer visits servers and regions using simply the methods *Enter*_object and *Leave*_object. An automatically generated report can be displayed at the end of the simulation. Furthermore, it is very easy to implement replications and confidence interval computation.

The class *Demos* is a discrete simulation package written by G. Birtwistle in 1979 at the University of Bradford. It may be considered as a large extension of the class *Gpsss*, for more details: *www.iro.umontreal.ca/~ vaucher/DEMOS/*.

DISCO is a combined discrete-continuous simulation package written by Keld Helsgaun, Roskilde University Center, Denmark in 1980. The class *Combinedsimulation* is a subclass of *Simulation*, it is designed to facilitate description and simulation of systems involving both discrete and continuous phenomena. The continuous part of a system is defined by differential equations implemented by objects *Variable* each one possessing two attributes: *State* and *Rate*. The class *Disco* derives from *Combinedsimulation* and includes methods to solve differential equations. The class *Pdeone* extends the class *Disco* with facilities for simulation of systems involving partial differential equations.

Complex System Modelling

SIMULA is largely used in many simulation domains: manufacturing systems, ecology, logistics, hospitals... In ecology (Breckling 1998) designed models of trees, roots, rodents, ground beetles and groups of lake fishes in interaction. The main simulation problem concerns the large number of objects and event instances. So the event management was modified to speed up the simulation.

An open library for manufacturing systems (Gourgand and Tanguy 1993, 1995) was designed to allow automatic generation of models thanks to a graphic tool. It contains *Conveyor*, *Machine*, *Storage* and other objects. The generator of the prefixed block, defining the model, was easily and quickly realized.

Recently, we used the class *Gpsss* to design hospital logistics models (Gourgand *et al.* 2005, 2007) based on queueing networks. We aimed to validate QNAP2 and Witness models of the same system. That enables us to quickly invalidate and correct the last models.

It is difficult to design system models able to internally simulate the system or one of its parts. For instance a system uses a computer to simulate its evolution. So when the system model is simulated the internal model must be simulated too. This kind of nested simulation defines the reflective simulation (Kindler *et al.* 2001-2003). Flexible manufacturing systems (FMS), including automatically guided vehicles (AGV), may fall into deadlock because the capacity of buffers is limited and there is no other storage (Fig. 4). The deadlock problem was studied in (Kindler *et al*. 2003c) and a solution was proposed in a deterministic context. It is based on a reflective simulation model that forbids job entries and anticipates the FMS behaviour until all the jobs are processed and carried to the exit or a condition insures the completion of the treatments. This object-oriented simulation model is based on a simplified view of the general FMS one. It is hierarchically defined.

- The main programme collects information needed to realize the dynamic allocation of several arrays, the numbers of machines, jobs, the buffer capacity and the size of bills of material.
- The first subblock initializes the constant information such as the job input sequence and the bills of material. It contains the declaration of the *FMS* class and creates the first instance of the simulation model at the top level (external model at level 1). The class parameters inform the model about its level and its upper model that allows communication between models.
- The *FMS* class contains the local classes of the system entities, mainly *Machine*, *Vehicle* and *Job*. The initialization part and the simulation kernel are defined in the life rules coroutine of *FMS* class.

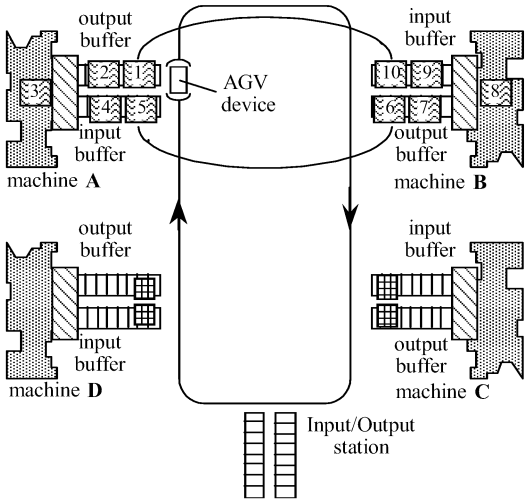


Figure 4: Constraints and Deadlock in FMS

When a *FMS* class instance is created, it starts its coroutine that initializes the external model (level 1) or makes a copy of the upper model state attributes into the nested internal model (level > 1). The selected vehicle move is prepared then the internal simulation starts and runs until a decision has to be tested. Then each decision is anticipated using a new nested internal model (level+1) which reports its best makespan and terminates. One could modify the model behaviour in order to limit the internal simulation time applying a partial search on the decision tree. Many errors may occur in the design of a reflective simulation model. The automatic generation of the global model, from the external one, was foreseen. It was demonstrated (Krivy *et al*. 2002) that only SIMULA is able to offer a suitable safety level, to facilitate the automatic generation and to manage multiple nested time axes.

CONCLUSION

Despite of its age SIMULA includes sophisticated and modern concepts. A lot of powerful extensions have been designed and compilers are available for the most popular materials and operating systems.
The automatic generation of SIMULA models is very easy when using graphic specification tools or simple files.
The model prototyping is very quickly realized by means of existing classes such as Gpsss and can be used to validate models built with dark graphic simulation tools.
The implementation of reflective simulation models is largely facilitated by SIMULA and it can serve to design more complex models.

REFERENCES

Backus J. W. et al. 1960."Report on the Algorithmic Language ALGOL60". *Numerische Mathematik 2*.

Birtwistle G. M. 1979. *A System for Discrete Event Modelling on SIMULA (Demos)*. Macmillan Press.

Breckling B. 1998. "The Application of the Object Oriented Programming Paradigm to Study Ecological Processes". In *24 th Conference ofthe ASU*. Salzau, Germany.

Dahl O. J. and K. Nygaard. 1965. *SIMULA - A Language for Programming and Description of Discrete Event Systems*. Introduction and User's Manual. Norwegian Computing Center, Oslo.

Dahl O. J. and K. Nygaard. 1966. "SIMULA - An Algol-based Simulation Language". *Communication of the ACM*,No. 9.

Dahl O. J. et al. 1968. *Programming Languages*. F. Genuys (ed), Academic Press, London.

Gourgand M. and A. Tanguy. 1993. "A SIMULA Object-Oriented Modelling Environment". In *European Simulation Symposium*, October 1993, Delft, The Netherlands.

Gourgand M., P. Kellert, A. Tanguy. 1995. "An Object Oriented Environment Based On a Transaction Approach and Simulation or Analytic Solvers". In *IMACS European Meeting*, August 1995, Gyor, Hungary.

Gourgand M., F. Mebrek and A. Tanguy. 2005. "Hospital Logistic Modelling and Simulation Case Study: Brancardage". In *ESM05*, 24-26 October, Porto, Portugal.

Gourgand M., F. Mebrek and A. Tanguy. 2007. "Modelling and Simulation of the Pharmacy of a New Hospital". In *ECSM07*, 4-6 June, Prague, Czech Republic.

Kindler E. 1995. "Tutorial on Mejtski's Diagrams". In *21st Conference of the ASU*, Lund, Sweden.

Kindler E.; I. Křivý and A. Tanguy. 2001. "Reflective simulation of discrete logistic and production systems". In *Modelling and Simulation, ESM 2001*, Prague, 6-9 June, 2001. 861-865.

Kindler E.; I. Křivý and A. Tanguy. 2002. "Reflective simulation of logistic and production systems". In *International Workshop HMS, MAS 2002*, Genoa, 97-102

Kindler E., I. Krivy, A. Tanguy. 2003a. "Systematics of Nested Simulation". In *International Workshop HMS 2003*. 18-20 September, Riga, Latvia. 191-195

Kindler E., I. Krivy, A. Tanguy. 2003b. "Simulation of Simulating Computerized Systems". In *International Workshop on Applied Simulation, MAS 2003*, 2-4 October, Bergeggi, Italy

Kindler E., I. Krivy, P. Lacomme, A. Tanguy. 2003c. "Simulation of FMS Including Automated Guided Vehicle". In *ESMC 2003*, 27-29 October, Naples, Italy, 122-126

Knuth D. E. 1969. *The Art of Computer Programming*. Addison-Wesley.

Křivý, I.; E. Kindler and A. Tanguy. 2002. "Software for Simulation of Anticipatory Production Systems". *International Journal of Computing Anticipatory Systems*, Vol. 11, 320-335

Sklenar J. 1997. "Introduction to OOP in SIMULA". URL http://staff.um.edu.mt/jskl1/talk.html

SSG SIMULA Standard Group. 1989. *SIMULA Standard 198 6* SIMULA a.s., Oslo, Norway.

SCICOS MODELICA

Thermalhydraulics applications developed in Scicos

Masoud Najafi*, Daniel Bouskela†

KEYWORDS
Simulation software, Thermalhydraulics, Modeling, Modelica, Scicos

ABSTRACT

The open-source hybrid simulation software Scicos has been extended to simulate models developed in the Modelica language. The Modelica language provides facilities to model physical systems at the component level. In this paper we present the thermalhydraulics library toolbox of Scicos and an industrial application with medium complexity developed and simulated in Scicos.

INTRODUCTION

Scilab (`www.scilab.org`) is an open-source software for scientific computing which is developed by INRIA. Scicos (`www.scicos.org`) is a toolbox of Scilab that provides an environment for the modeling and simulation of dynamical systems (Bunks et al. 1999, Chancelier et al. 2002, Campbell et al. 2005). Scilab and Scicos can be respectively compared with Matlab and Simulink. The underlying hybrid formalism in Scicos allows the modeling of very general dynamical systems: systems including continuous, discrete-time and event based behaviors. Scicos, originally based on causal system modeling, uses causal modules with explicit input and outputs. In order to implement a causal module, explicit blocks with explicit inputs and outputs (standard Scicos blocks) can be used. Scicos has been extended to simulate component based models (Najafi et al. 2005; 2004). Implementing this new feature in Scicos needed to use a new block type, *i.e.*, *Implicit blocks*. Thus, in Scicos, two block types can be used: explicit and implicit blocks.

Implicit blocks, or components, have implicit ports. An implicit port is different from an input or an output port in the sense that connecting two such ports imposes a constraint on the values at these ports, but does not imply the transfer of information in an *a priori* known direction as it is the case when an output port is connected to an input port. In order to be able to use implicit blocks in addition to explicit ones in Scicos, several new features have been added to Scicos. Implicit blocks or components are interfaced via special connection links associated with physical quantities such as current or voltage in electronics, or flow or pressure in hydraulics. Contrary to explicit blocks whose behaviors are written in C or in Fortran, implicit blocks cannot be modeled as black box objects; the equations that describe the behavior of an implicit block must be available to the compiler for the generation of explicit code. In order to describe the behavior of implicit blocks and the construction of diagrams based on these blocks, the *Modelica language* has been adopted in Scicos.

Modelica is a modern object-oriented declarative language based on equations instead of assignment statements. Modelica has multi-domain modeling capability, *e.g.*, electrical, mechanical, thermodynamics, hydraulics, and control systems can be described in Modelica (Fritzson 2004). The Modelica language is independent of simulation tools and is developed by the Modelica association (`www.modelica.org`).

With Modelica, in order to write a complicated model easily and efficiently, the model can be decomposed into several components. Each component can be in turn decomposed into other sub-components, or be directly described using Modelica text. Then, by interconnecting these components, the model can be constructed. In each Modelica program, first, the parameters and the variables are declared; then the equations which define the behavior of the model are written. In Modelica, equations are composed of expressions both on the left hand side and the right hand side. It is neither required to write the equations in form of assignments, nor to write the equations in a specified order.

In the following sections, we detail the thermalhydraulics library components developed for Scicos and an industrial application that uses the library. This application is taken from the field of power plants for the generation of electricity.

THERMALHYDRAULICS TOOLBOX FOR POWER PLANTS

EDF has developed a nearly complete Modelica thermalhydraulics library for power plants (see an overview description in (Avenas et al. 2004)). This library has been tested with *Dymola* (`www.dynasim.com`). Since the Modelica models can be simulated independently from the simulation tool, this library is used as the foundation for an open source thermalhydraulics toolbox for Scicos. The library supports 1D one-phase flow, *i.e.*, water or steam, and two-phase flow, *i.e.*, water/steam.

*masoud.najafi@inria.fr INRIA, Rocquencourt, Domaine de Voluceau, 78153, Le Chesnay, France

†daniel.bouskela@edf.fr, Électricité de France (EDF), 6 Quai Watier, Chatou, 78401, France

Furthermore, the components handle flow reversal and the zero flow case. The numerical scheme used for the library is the finite volume method with staggered grid scheme. This method is known as a stable and precise. The whole hydraulic network is a linear 1D assembly of 0D control volumes. The values of pressures, temperatures, and enthalpies are calculated at the center of the control volumes; the flows are calculated at the edges of the control volumes. There are two grid types:

- Volumes (or nodes): they are superimposed on the control volumes. They implement the energy and the mass balance equations. So they do not compute pressure losses.
- Two-ports elements (or edges): they implement the momentum balance equations. They are thus necessarily adiabatic, (*i.e.*, they do not exchange energy with the outside). Volumes are interconnected via two-port elements.

The thermalhydraulics models are obtained by interlacing volumes and two-ports elements. Volumes are multiports (*i.e.*, they can have more or less than two ports). In theory, it is possible to directly connect together two two-ports elements or two volumes. In doing so, we obtain an element which is the sum of the two connected elements. But, when connecting two elements that contains state variables (*i.e.*, differential variables), this operation leads to a high index DAE problem (Campbell 1994). In fact, such connections impose additional algebraic relations on differential variables. High index problems require applying index reduction methods to be able to simulate the model with standard solvers. Index reduction can be performed automatically, but the problem of finding the consistent initial conditions is difficult to solve automatically, *i.e.,* what are the extra initial conditions that must be removed ?. So in practice, direct connection of two identical elements is only possible if at least one of the elements is static (*i.e.,* without differential variables). For example, it is always possible to connect a two-ports element to a valve, because the latter is most often considered as a zero volume (or singular) element and is therefore always static.
In developing the models, we have used some but not all of the features of the Modelica language. With Modelica, continuous-time and discrete-time equations, internal and external algorithms and stateless functions can be used. In principle, discrete-time equations should not be used in thermalhydraulics applications, because thermalhydraulics is continuum physics (it belongs to the continuum media mechanics category). However it is often more efficient (in terms of computing time) to model very fast dynamics by replacing continuous-time states by discrete-time states. This often happens within the mechanical parts of the system, such as check valves, which may be modeled as simple switches for most applications. The external/internal stateless functions are useful only to describe utility functions, *e.g.,* the physical properties of the fluids.
Modelica is an object oriented language, so models can be structured using decomposition, *i.e.,* breaking the model into sub-models and connecting them via ports, and using inheritance, *i.e.,* defining base models with generic properties and building the models by inheriting the properties of these base models. But, using the Modelica inheritance mechanism to model thermalhydraulics is difficult. The reason lays in the fact that inheritance is designed to be used on discrete objects. But in thermalhydraulics, objects only come up after space disretization with the finite volume method, *i.e.,* as aforesaid, when the system model is broken into a linear assembly of nodes and edges, which are not natural (technological) objects. On the other hand, the inheritance mechanism is useful to develop the mechanical parts of the system, such as the mechanical part of the valves, which are natural objects.
The EDF thermalhydraulics library has been developed and tested with Dymola, which covers most features of the Modelica language. However, only the essential subsets of the language have been used: the continuous-time equations, a minimum support for the discrete-time behavior, and the structured objects without heritage. Thus, the current version of the Modelica compiler of Scicos which supports these features, can be used to exploit the library. The current thermalhydraulics library toolbox of Scicos, which up to now only implements a small fraction of the full EDF library, is given in Fig. 1.

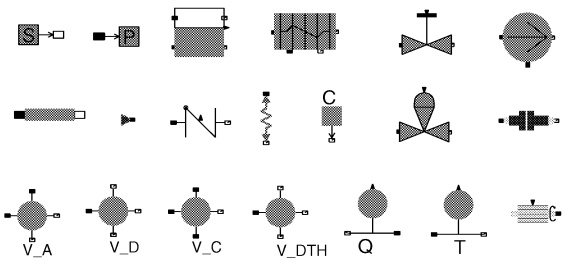


Figure 1: Thermalhydraulics toolbox in Scicos

IAPWS-IF97 library

The IAPWS-IF97 standard library that gives the properties of water and steam in different pressure and temperature domains has been used. The IAPWS Industrial-Formulation (`http://www.iapws.org`) 1997 consists of a set of equations for different regions which cover the following range of validity:

$$273.15°K \leq T \leq 1073.15°K, \qquad P \leq 100 \text{ Mpa}$$
$$1073.15°K < T \leq 2273.15°K, \qquad P \leq 10 \text{ Mpa}$$

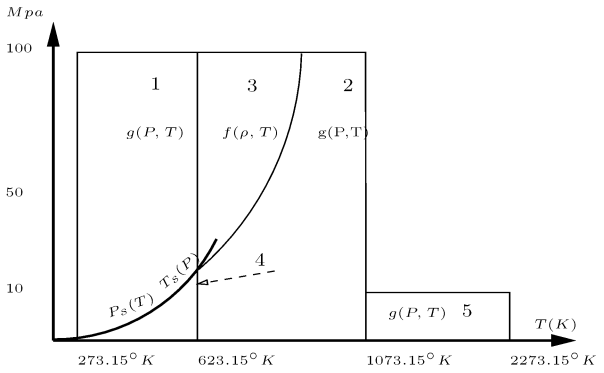


Figure 2: Structure and regions of IAPWS-IF97 library

The Fig. 2 shows the five regions into which the entire range of validity of IAPWS-IF97 is divided.

- region 1 for the liquid state from low to high pressures,
- region 2 for the vapor and ideal gas state,
- region 3 for the thermodynamic state around the critical point,
- region 4 for the saturation curve (vapor-liquid equilibrium),
- region 5 for high temperatures above 1073.15°K and pressures up to 10 MPa (100 bar).

Regions 1, 2, and 5 are both individually covered by a fundamental equation for the specific Gibbs energy $g(P,T)$, region 3 by a fundamental equation for the specific Helmholtz energy $f(\rho,T)$, and the saturation curve, corresponding to region 4, by a saturation-pressure equation $P_s(T)$. All thermodynamic properties can then be calculated from these fundamental equations by using the appropriate thermodynamic relations. For region 4 a saturation-pressure equation has been developed.
On the basis of IAPWS-IF97, a package for the calculation of the water and steam properties was provided by EDF. This package is a modified version of the original Modelica implementation of H. Tummescheit. It includes tables and functions to access all properties. The IF97 region can be computed automatically. In order to simplify the numerical resolution of algebraic loops (in particular for the computation of the initial state), the IF97 region can also be set manually by the user to a fixed value if it does not change (*i.e.,* if there is no region crossing) during simulation (no phase transition for instance).

AN INDUSTRIAL APPLICATION: A POWER PLANT COOLING SYSTEM

As a simple industrial application, we have modeled an intermediate system used to cool large auxiliary equipments located in the conventional part of nuclear power plants.
The system includes a heat source to account for the heat generated by the equipments, a pumping station, a cooling station, and a controller, see Fig. 3.

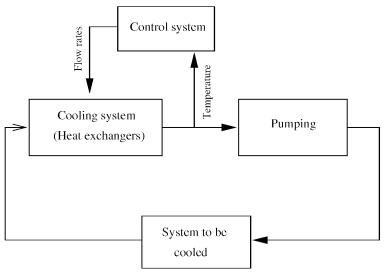


Figure 3: Structure of the cooling system

The Scicos model of the cooling system is given in Fig. 4. The equipments to be cooled down have been modeled with a heat source block. The heat source is connected to the constant volume chamber with an additional thermal input port to heat up the fluid. The heat transfer from the heat source to the chamber is modeled with a heat flux block. The pumping station consists of three parallel centrifugal pumps. Under normal operating conditions, two pumps are working and one pump is stopped. The latter is used as a backup pump in case of the failure of one of the two other pumps. Each pump is powered by a DC motor. The pressure loss in the circuit due to the pipe network is modeled with a single control valve. Another control valve is used to model a possible leakage in the circuit. The temperature sensor block measures the temperature of the water at the outlet of the cooling station.
The objective of the system is to cool the equipments with the two following constraints: the temperature at the outlet of the cooling station should always be 17°C, and the water flow rate through the cooling station should be quasi-constant, whatever the amount of heat to be evacuated. To that end, the cooling station comprises two heat exchangers and a bypass in parallel. The flow through each branch is controlled by a control valve. The positions of the control valves are correlated in order to keep the flow rate quasi-constant at the outlet of the cooling station: when the water temperature at the inlet rises (resp. decreases), the heat exchangers control valves open (resp. close), and the bypass valve closes (resp. opens). The temperature at the outlet is kept to the setpoint with a closed loop PI controller, that delivers a single signal used by the 3 control valves. The controller is developed with standard Scicos blocks. After compilation and model simplification, the model equation system is reduced to a DAE with 157 state variables.
The initial conditions are the following: the water temperature is 290°K, the temperature setpoint is

290.15°K. Two pump motors are running at nominal speed, the third pump motor is stopped. For simplicity, in this model three pumps are always active. The heat source is constant with a temperature of 323.15°K. The transient at the beginning of the simulation is due to the fact that the system does not start from steady state. The simulation results for four system variables, *i.e.,* water flow rates in the two branches of the cooling station, the water temperature at the cooling station outlet, and the water flow rate in the main loop are given in Fig. 5.

PERSPECTIVES OF MODELICA IN SCICOS

The Modelica models presented in this paper are compiled with Modelicac (acronym of the Modelica Compiler). Modelicac, developed by Imagine (www.amesim.com), is an open source compiler for a subset of the Modelica language. This compiler comes with a module that generates C codes for Scicos targets, but since it is open source and standalone, code for other targets can also be generated. Modelicac is the result of the SIMPA[1] project with the participation of INRIA, IMAGINE, EDF, IFP, and Cril Technology. This compiler supports most continuous-time features of Modelica and provides a minimum support for the discrete-time part. The main features of the compiler are the simplification of the Modelica models and the generation index-1 DAE ready for simulation. It supports discontinuous model switching, and provides the analytical Jacobian of the model. It does not support inheritance, nor higher index DAEs.
Recently the ANR[2]/RNTL SIMPA2 project has been launched to develop a more advanced Modelica compiler. The main objectives of this project are to extend the SIMPA compiler to fully support inheritance and hybrid systems, and give the possibility to solve inverse problems by model inversion for static and dynamic systems. Static model inversion will be used to find the initial state from initial conditions defined as an inverse problem: the internal states and the inputs will be computed from the outputs of the model. Dynamic model inversion will be used *e.g.,* for the sizing of the actuation chain in mechatronics systems.

CONCLUSION

The current thermalhydraulics toolbox of Scicos was presented. Using this toolbox, the user can develop thermalhydraulics models of small/medium complexity by interconnecting basic blocks such as pressure sources, pipes, valves, etc. The user can also develop new components and include them in the toolbox. A few of the Scicos models used to validate the library components was described. In the last section of the paper, the model of a cooling system with medium complexity using the Scicos thermalhydraulics library was given as an industrial example. Future work will be to develop a more complete thermalhydraulics library for Scicos, in order to be able to model and simulate real industrial applications.

[1]Simulation pour le Procédé et l'Automatique
[2]French National Research Agency

REFERENCES

Avenas C.; Bouskela D.; and et al., 2004. *Quasi-2D Steam Generator Modelling with Modelica. Conference ISC'2004, Malaga, Spain.*

Bunks C.; Chancelier J.P.; Delebecque F.; Gomez C.; Goursat M.; Nikoukhah R.; and Steer S., 1999. *Engineering and Scientific Computing with Scilab.* Barkhausen, Le Chesnay, France, 1st ed. ISBN 0817640096.

Campbell S.L., 1994. *Numerical Methods for Unstructured Higher Index DAEs. Annals of Numerical Mathematics*, 1, 265–278.

Campbell S.L.; Chancelier J.P.; and Nikoukhah R., 2005. *Modeling and simulation Scilab/Scicos.* Springer Verlag, 1st ed.

Chancelier J.P.; Delebecque F.; Gomez C.; Goursat M.; Nikoukhah R.; and Steer S., 2002. *An introduction to Scilab.* Springer Verlag, Le Chesnay, France, 1st ed. ISBN 2-287-59731-X.

Fritzson P., 2004. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1.* Wiley-IEEE Press, 1st ed. ISBN 0-471-47163-1.

Najafi M.; Azil A.; and Nikoukhah R., 2004. *Extending scicos from system to component level simulation. ESMC2004 international conference, Paris, France.*

Najafi M.; Nikoukhah R.; Steer S.; and Furic S., 2005. *New features and new challenges in modeling and simulation in Scicos. IEEE conference on control application, Toronto, Canada.*

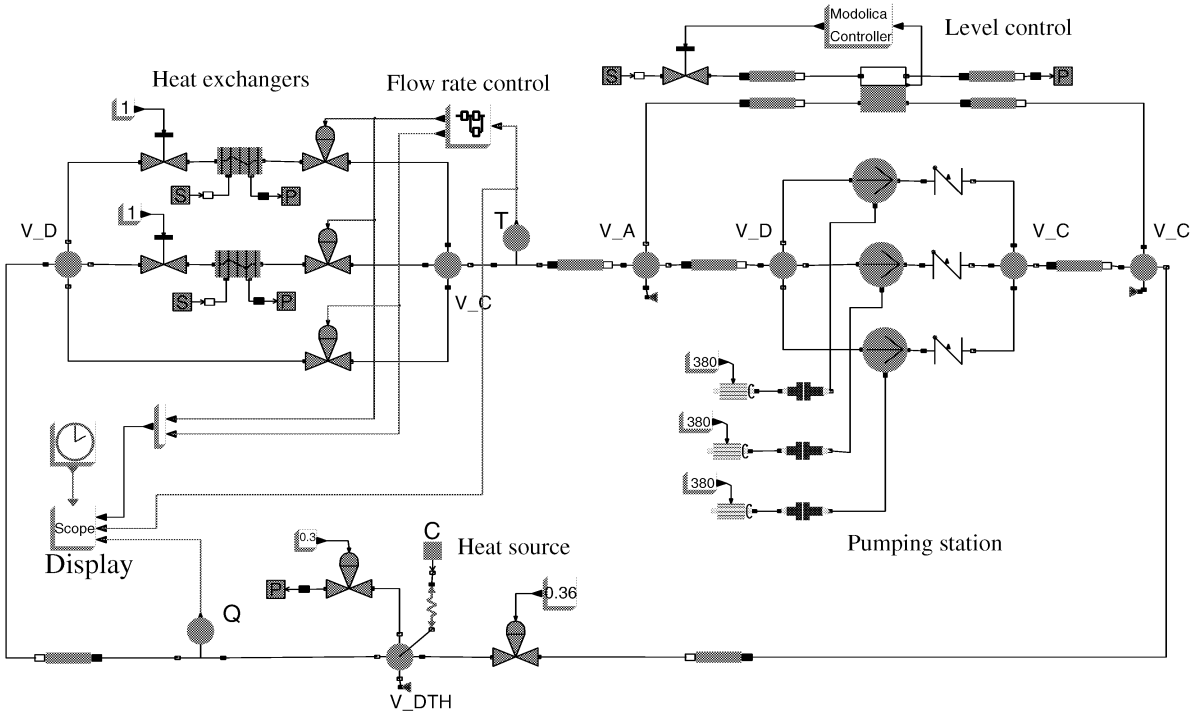


Figure 4: A simplified cooling system modeled in Scicos

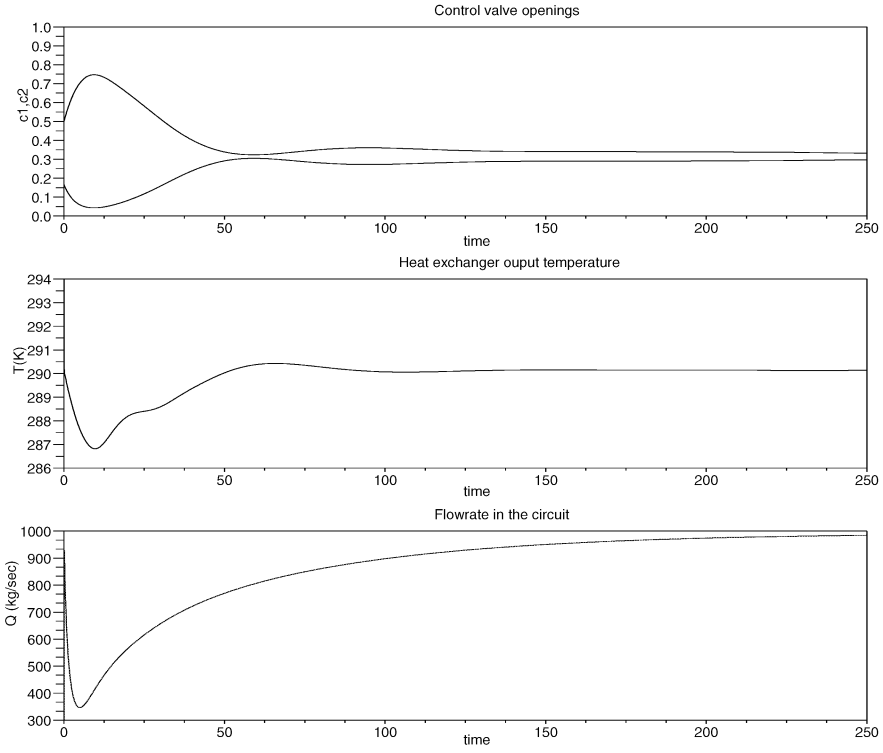


Figure 5: Simulation Result of the Model of Fig. 4.

Modeling complex systems with Modelica in Scicos: Application to mean value spark engine

Masoud Najafi[*] Zakia Benjelloun-Dabaghi[†]

KEYWORDS
Spark-ignited engine, Dynamic system simulation, Simulation software, Scicos; Modelica

ABSTRACT

This paper presents modeling and simulation of a spark-ignited (SI) engine in the Scicos software. The open-source hybrid simulation software Scicos has been extended to partially support the Modelica language. Modelica is a programming language for modeling dynamical systems at component level. In this paper, after a brief review of the physical processes inside a spark-ignited engine, we use the Modelica language to model the different parts of the engine. Then after, we implement the model of an SI engine in order to implement an idle speed control system.

INTRODUCTION

Scilab (`www.scilab.org`) is a free and open-source software for scientific computing which has been developed in INRIA. Scicos (`www.scicos.org`) is a toolbox of Scilab and provides an environment for modeling and simulation of dynamical systems (Campbell et al. 2005). They can be compared with Matlab and Simulink respectively. The underlying hybrid formalism in Scicos allows modeling very general dynamical systems: systems including continuous, discrete-time and event based behaviors which is generally referred to as hybrid systems.
Scicos is originally based on system level modeling. In order to implement a model with this approach, causal modules(*i.e.,* modules with explicit inputs and outputs) can be used. Scicos has been developed to simulate component based models. In this approach Scicos uses Implicit blocks which are also referred to as components. Implicit blocks are connected to each other via special links. This links are associated with physical quantities such as current or voltage in electronics, or, flow or pressure in hydraulics. Contrary to explicit block whose behaviors are written in C or Fortran languages, implicit block cannot be modeled as black box objects with inputs and outputs; the equations realizing the behavior of an implicit block must be available to the compiler for generation of a causal code. In order to describe the behavior of implicit blocks and the construction of diagrams based on these blocks, the *Modelica language* are used in Scicos.

Modelica language

Modelica is a modern object-oriented programing language based on equations instead of assignment statements. Modelica has a multi domain modeling capability, *e.g.*, electrical, mechanical, thermal-hydraulic, hydraulic, and control systems can be described by Modelica (Fritzson 2004). The Modelica language is independent of the simulation tool and has been developed by the Modelica association (`www.modelica.org`).
In Modelica, in order to write a complex model easily and efficiently, the model is decomposed into several components. Each component which can also be composed of other subcomponents is modeled in Modelica. Then, by interconnecting these components the model can be constructed (Najafi 2005).
In the following sections, first we explain the SI engine components that have been developed at IFP with the Modelica language in Scicos. Then, the model of a mean value SI engine is assembled by interconnecting the components in order to implement an idle speed control systems. Scicos generates a Modelica program for this model and then simulate it.

MEAN VALUE SI ENGINE

The model of the SI engine described in this paper is a nonlinear, low frequency of a fuel-injected four cylinders SI engine which is generally referred to as a *mean value* model. Mean value engine models attempts to capture dynamics in a time-scale spanning over several combustion cycles. Fast events are not of interest other than their effects on a larger scale. Most cyclic dynamics are modeled by their average value over a cycle. The speed and torque output of the engine and the pressure in the inlet manifold are the aspects of most interest in mean value engine model that we have developed. Mean value engine model generally represents a basis for the development of different engine control strategies.
The model of the overall engine is composed of several subsystems or components. In order to develop

[*]masoud.najafi@inria.fr, INRIA-Rocquencourt, Domaine de Voluceau, 78153, Le Chesnay, France. www.inria.fr
[†]Zakia.benjelloun-dabaghi@ifp.fr, IFP (French Institute of Petroleum), 1 Avenue Bois-Préau, 92852, Rueil-Malmaison, France. www.ifp.fr

the model of the SI engine easier, the engine subsystems are modeled separately. The elements of the system including the air throttle, the intake manifold, exhaust gas recirculation (EGR), the canister purge mechanism, sensor dynamics, combustion chamber, and the load and speed perturbations are modeled. Inherent system delays in the four-stroke engine cycle including the induction-to-power stroke delay, effects of the air/fuel ratio or fuel richness are not modeled in this paper. The system including fundamental components, sensors, and actuators is illustrated in Fig. 1.

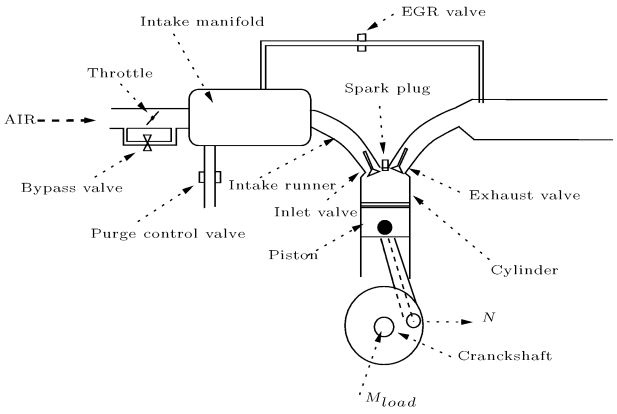


Figure 1: Principle sketch of SI-engine

In the following subsections, a brief description of the engine components corresponding the physical phenomena used in the model will be given (Silverlind 2001). These components are shown in Fig. 2.

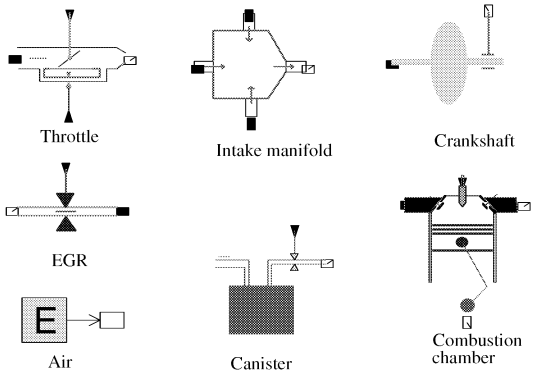


Figure 2: Scicos toolbox for engine components

Air intake throttle

The air throttle controls the air flow rate into the air manifold and the combustion chamber. The air throttle can be modeled as a flow restriction. The model of a flow restriction highly depends on the pressure difference across the restriction, if small enough, the gas density is considered equal on both sides, *i.e.,* the gas is considered as an incompressible fluid. If, on the other hand, large pressure differences can be expected the restriction should be modeled assuming compressible fluids. Thus, the intake air mass flow rate ($\dot{M}_{at}$) is modeled with equations (1).

$$\dot{M}_{at} = (\alpha + k\alpha_0)\frac{P_{at}}{\sqrt{R_{at}T_{at}}}\Psi(\frac{P_{man}}{P_{at}})^{0.5} \qquad (1)$$

where α and α_0 are the throttle plate open area and bypass air passage, respectively. P_{man}, P_{at}, R_{at}, and T_{at} are the manifold's pressure, the environment air pressure, the air specific gas constant, and the environment air temperature, respectively. $\Psi(x)$ is a scalar function defined as:

$$\Psi(x) = \begin{cases} \frac{2\gamma}{\gamma-1}(x^{\frac{2}{\gamma}} - x^{\frac{\gamma+1}{\gamma}}) & \text{if } x \geq (\frac{2}{\gamma+1})^{\frac{\gamma}{\gamma-1}} \\ \gamma(\frac{2}{\gamma+1})^{\frac{\gamma+1}{\gamma-1}} & \text{otherwise} \end{cases} \qquad (2)$$

where $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heats of the air. Note that we have also assumed that there is no back flow and the temperature is unchanged across the throttle. Interested readers are referred to (Nakayama and Boucher 1999) for more details.

When the engine is in idle mode, *i.e.,* $\alpha = 0$, the necessary air for the maintaining the minimum power of the engine is supplied through an air passage, called air bypass passage. The bypass area (α_0) is controlled by the engine control unit (ECU) to maintain the necessary air flow during idle speed control.

The schematic model of the throttle block in Scicos is given in Fig. 1. The air throttle component has two implicit ports and two explicit inputs. In Fig. 2 the square ports are implicit and triangle ones are explicit. Implicit ports represent inlet and outlet air flows and explicit input ports represent control signals for α_0 and α. The implicit ports, modeled with the `connector` clause in Modelica, caring physical variables, *i.e.,* pressure (P), mass flow rate ($\dot{M}$), temperature (T).

Exhaust gas recirculation (EGR)

In order to reduce harmful emissions resulting form the combustion, some of the exhaust gas is diverted back into the combustion process. In this method the inlet and exhaust manifolds are connected with a pipe and the recirculated gas flow rate is controlled by a valve (Silverlind 2001). The total gas mass flow through the EGR control valve, ignoring the back flow, is obtained using the same formula used for the throttle model. The schematic model of the EGR block in Scicos is given in Fig. 2. The EGR block has two implicit ports and one explicit input port as the control signal for the EGR control valve opening.

Canister

Most of the HC emissions in modern cars are from the exhaust, but a considerable part also comes from evaporative losses in the fuel tank. Most modern cars use an evaporative emissions management system to reduce these emissions. The basic function of this system is to trap and store the fuel vapors from the fuel tank in a canister until the engine is started. Then after the trapped fuel vapors is drawn into the engine by intake air manifold and combusted. To control the flow of vapors into the engine, a purge control valve with no back flow is used. Similar to the model of the EGR system, the mass flow rate of the air/fuel mixture through the canister purge is modeled as a restriction. The schematic model of the Canister block in Scicos is given in Fig. 2.

Intake Manifold

The air passed through the air throttle, the EGR, and the canister are mixed in the intake manifold and are send into the combustion chamber through the intake runner. We have assumed an isothermal manifold heat transfer, *i.e.,* constant manifold air temperature.
The air in the intake manifold is composed of fresh air, fuel, and burnt gas. Assuming that the concentration of fresh air (Y_{air}) is mush bigger that of the burnt gas (Y_{ex}) and fuel (Y_f), *i.e.,* $\{Y_{ex}, Y_f\} << Y_{air}$, we can define the relative mass concentrations in the intake manifold. The concentrations can be described as functions of the partial pressures of fuel and air in the intake manifold, *i.e.,*

$$\begin{aligned} M_{man}\dot{Y}_{ex} &= \dot{M}_{ex} - Y_{ex}\dot{M}_{run} \\ M_{man}\dot{Y}_f &= C_f\dot{M}_{can} - Y_f\dot{M}_{run} \\ Y_{air} &= 1 - Y_b - Y_f \end{aligned} \tag{3}$$

where M_{man} and $\dot{M}_{run}$ are the total mass of gases in the intake manifold and mass flow rate through the intake runner, respectively. C_f is the fuel concentration in the canister vapors. Using the ideal gas law, we can obtain the following model for the intake manifold pressure.

$$\begin{aligned} M_{man}R_{air}T_{man} &= P_{man}V_{man} \\ \dot{P}_{man}V_{man} &= R_{air}N_{air}(\dot{M}_{air}T_{air} + \dot{M}_{ex}T_{ex} + \\ &\quad \dot{M}_{can}T_{can} - \dot{M}_{run}T_{man}) \\ \dot{M}_{man} &= \dot{M}_{air} + \dot{M}_{ex} + \dot{M}_{can} - \dot{M}_{run} \end{aligned} \tag{4}$$

In SI engines, the inlet manifold pressure is reduced by the throttle in order to control the output torque. As will be seen later on, the flow rate in the intake runner ($\dot{M}_{run}$) is imposed by the pumping mechanism of the combustion chamber and the crankshaft rotation.
The schematic model of the manifold block in Scicos is given in Fig. 2. The block has four implicit ports and one explicit output port representing the MAP (Manifold Air Pressure) sensor.

Combustion chamber

The combustion chamber is the heart of the engine. The air/fuel mixture flows into the cylinders and reacts and usable energy is extracted from the heated gas which is then expelled. In an SI engine, the air is mixed with fuel before entering the cylinders. In this paper, the effects of the air/fuel ratio are not modeled. The cylinder is continuously swept by a piston which is connected to the crankshaft via a rod. The top of the cylinder houses intake and exhaust ports and a spark-plug in SI engines. The cylinder and the crankshaft have two important roles: torque generation and air pumping. When gas burns and expands, the piston is forced down. The downward movement is then transformed into rotational movement. The applied torque on the crankshaft depends on several parameters, such as the air/fuel mixture ratio, spark ignition time, manifold pressure, angular velocity of the crankshaft, etc. Since there is no accurate and simple physical model describing the generated torque, it is quite often that a map is used. This map gives the optimum generated torque as a function of the manifold pressure (P_{man}), and the angular velocity of the crankshaft (ω). Thus, the optimal obtainable torque is defined as

$$\tau_{gen}^{opt} = F(\omega, P_{man}).$$

where τ_{gen}^{opt} is the optimum deliverable torque by the power stroke. This map gives the value of the produced torque regardless of other important effects such as the effects of spark advance.
Adjusting the spark advance timing, we can optimize engine efficiency to deliver peak combustion pressure when the piston reaches about 10° after top dead center (TDC) angle. Incorrect spark timing can have a significant effect on emission output and vehicle drivability. The amount of the spark advance needed by the engine varies as function of the number of different operating conditions. The coolant temperature, fuel quality, and engine load are just a few of the many factors that can significantly impact ideal ignition time (Heywood 1988, Aquino 1981, Degobert 1995). The effects of the spark timing on the produced torque is obtained by a using a experimentally obtained map. The map that we have used in our simulation gives the spark advance efficiency or the ratio of the produced torque with respect to the optimal torque, *i.e.,*

$$\eta = \frac{\tau_{gen}}{\tau_{gen}^{opt}} = H(|SA|)$$

where $|SA|$ is the absolute value of the spark advance timing. Note that $H(0) = 1$ and $|SA| < 40°$.
The up/down movement of the cylinder creates a pumping effect; when the piston moves downward, the air is

inhaled from the intake manifold and when the piston moves upward, the burnt air is exhaled to the exhaust manifold. In an internal combustion engine the pressure on the intake side will normally be lower than on the exhaust side. Pumping gas from low to high pressure costs energy and this energy is taken from the crankshaft. The amount of the pumped air which depends on several parameters such as the cylinder volume, the angular velocity of the crankshaft, pressure in the intake manifold, pressure in the exhaust manifold, and the air temperature is described with a map. This map describes the total gas flow rate as a function of manifold pressure P_{man} and engine speed ω, *i.e.,* $G(\omega, P_{man})$. It is well known that the air mass flow rate in the runner can be expressed as follows.

$$\dot{M}_{run} = G(\omega, P_{man}) \frac{\omega P_{ex} V_{cyl}}{T_{ex} R_{air}}.$$

The maps used in our model are obtained at IFP for a four cylinders SI engine. The schematic model of the combustion chamber in Scicos is given in Fig. 2. This block has three implicit ports for the air intake runner, the exhaust outlet, the connection with the crankshaft. The block has one explicit input port for spark advance signal coming from the controller.

Crankshaft dynamics and perturbations

The crankshaft dynamics are modeled using the Newton's second law for rotating masses as follows.

$$\begin{array}{lcl} J\dot{\omega} & = & \tau_{in} - \tau_l - \tau_{fric} \\ \tau_{in} & = & (1+\sigma_1)\tau_{gen} \\ \tau_l & = & \tau_{load} + \sigma_2 \\ \tau_{fric} & = & \kappa(\omega + \sigma_3) \end{array} \quad (5)$$

where τ_{in} is the total produced torque including all perturbations due to instabilities in combustion's, differences in generated torque in cylinders, and variations in fuel injection in different cylinders. The term σ_1 is a zero mean random noise, τ_l represents the load applied on the engine including controllable loads such as effects of A/C or anti-frost systems on the engine and uncontrollable perturbations modeled with a zero mean random noise. The term τ_{fric} represents the friction losses. The variations in the crankshaft speed is also modeled with a zero mean random noise σ_3 .
The schematic model of the crankshaft block in Scicos is given in Fig. 2. This block is connected to the combustion chamber block via an implicit port representing the mechanical connection of the crankshaft to the combustion chamber. The block has an explicit output port which provides the rotation speed (ω).

Intake manifold pressure sensor (MAP)

For all measurement systems, one desire that a sensor is fast compared to the system it monitors. If the sensor is fast, its dynamics can usually be neglected when modeling the system. In MAP-sensors used in cars, this is not the case and a first order model is used to estimate the manifold pressure, *i.e.,*.

$$k_{senor} \hat{P}_{man} = P_{man} - \hat{P}_{man}$$

where $\hat{P}_{man}$ is the measured value for P_{man} and k_{senor} is a time constant that should be identified.

APPLICATION: IDLE SI ENGINE SPEED CONTROL

In this section, the engine components are used to build the model of an SI engine. The engine model is then used to validate start-up and idle speed control strategies on the engine. The model of the engine is given in Fig. 3. The controller is developed with standard (explicit) Scicos blocks. Its modeling with explicit blocks in Scicos has the advantage of using the rich control toolbox of Scilab for controller design. For example, we can design PID, LQG controllers with Scilab and test their directly in Scicos.
In our model, the selected controller is relatively simple, *i.e.,* a PI controller. This controller will be active as soon as the engine speed exceeds 700 RPM. During the start-up phase, the spark advance is set to 20° and $\alpha_0 = 15\%$. When engine speed superseded the 700 RPM threshold, the control is handed over to the PI controller that adjusts the spark advance and the bypass area as a function of the reference speed, *i.e.,* 750 RPM and instantaneous MAP sensor and the engine speed. The results of a start-up from zero RPM and a idle speed control simulation is given in Fig .4. In this simulation different loads (τ_l) are applied at instants t=20 sec and t=40 sec. In the bottom plot of figures the load (τ_l) are given. In the top plot of Fig. 4 the engine speed is shown to be relatively regulated around 750 RPM in spite of the loads and random perturbations. The middle plot of Fig. 4, gives the intake manifold pressure that decreases from atmosphere pressure as engine starts up and varies as load changes.

Conclusion

In this paper we modeled a mean value SI engine with Modelica in Scicos. It should be noted that this model has been already modeled and simulated in Simulink at IFP. The modeling in Modelica was performed in order to compare two modeling environments. Modeling in Modelica has several advantages: Modelica is a declarative language with which very general hybrid systems can be modeled. The Modelica models are independent of the simulation tool and can be simulated in any Modelica simulator. In the Scicos simulator Modelica models are joint with implicit blocks. The overall model is then built by interconnecting the implicit blocks. Another important advantage of using Modelica lies in the

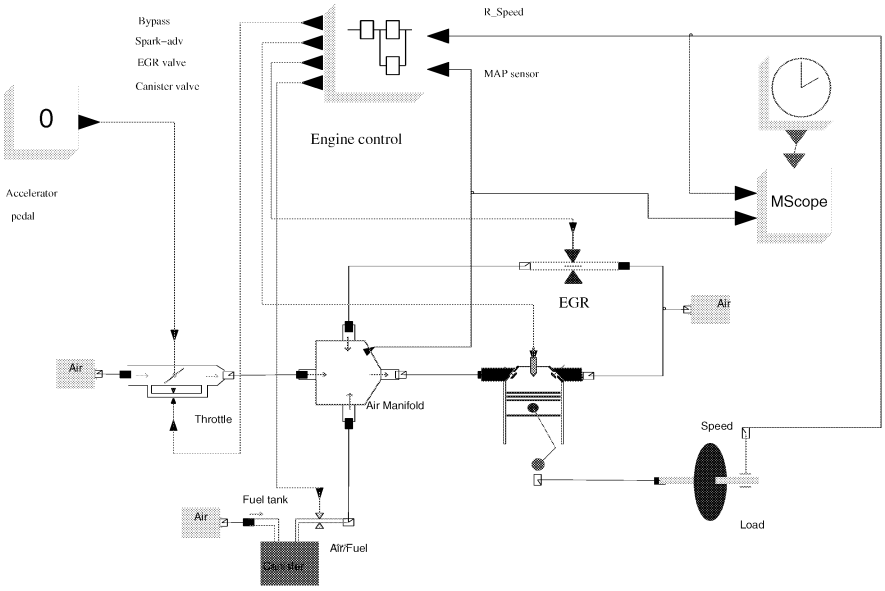


Figure 3: The Scicos model for a mean SI engine

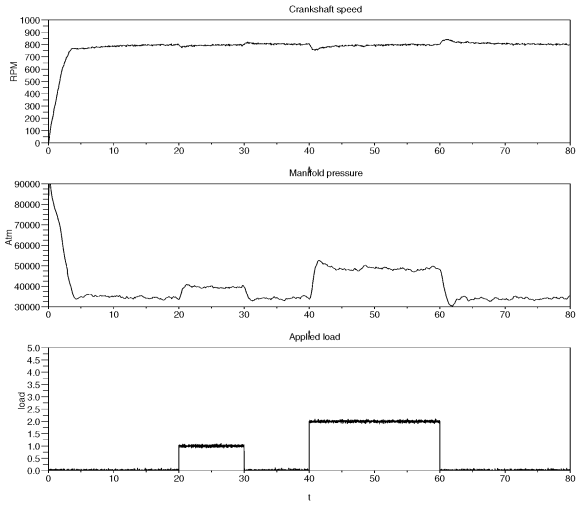


Figure 4: simulation result for start-up and idle speed control of the engine

symbolic manipulation of models. Because it gives the possibility of several simplifications such as efficient discontinuity handling, index reduction and generation of the analytical Jacobian.

REFERENCES

Aquino C., 1981. *Transient A/F control characteristics of the 5 liter central fuel injection engine.* In *SAE Technical Paper Series.*

Campbell S.L.; Chancelier J.P.; and Nikoukhah R., 2005. *Modeling and simulation Scilab/Scicos.* Springer Verlag, 1st ed.

Degobert P., 1995. *Automobiles and Pollution.* In *Society of Automotive Engineers, Inc. Translation of Automobile et pollution.*

Fritzson P., 2004. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1.* Wiley-IEEE Press, 1st ed. ISBN 0-471-47163-1.

Heywood J.B., 1988. *Internal Combustion Engine Fundamentals.* McGraw-Hill series in mechanical engineering, international edition.

Najafi M., 2005. *The Numerical Solver for the Simulation of the Hybrid Dynamical Systems.* Doctor of science thesis, Paris XII University.

Nakayama Y. and Boucher F., 1999. *Introduction to Fluid Mechanics. Arnold, London, Great Britain.*

Silverlind D., 2001. *Mean Value Engine Modeling with Modelica. M, Department of Electrical Engineering, Linkoping University*, Master's thesis.

IMPLEMENTATION IN MODELICA OF A VIRTUAL-LAB FOR TESTING WASHING MACHINE DESIGNS

Carla Martín-Villalba
Alfonso Urquia
Sebastián Dormido
Departamento de Informática y Automática, UNED
Juan del Rosal 16, 28040 Madrid, Spain
{carla, aurquia, sdormido}@dia.uned.es

Félix Martínez
Ingeniería Mecánica, IKERLAN
J. M. Arizmendiarrieta 2, 20500 Arrasate, Gipuzkoa, Spain
Felix.Martinez@ikerlan.es

KEYWORDS
Interactive simulation, object-oriented modeling, hybrid-models, industrial processes

ABSTRACT

The implementation of a virtual-lab for testing designs of drum-type washing machines is discussed. It is applied to the analysis of an industrial washing machine (120 Kg load capacity) manufactured by Fagor Industrial. The virtual-lab has been completely programmed using Modelica language. The washing machine dynamic model has been developed using *MultiBody* Modelica library. The interactive visual interface (i.e., the virtual-lab view) has been implemented using *VirtualLabBuilder* Modelica library. The virtual-lab supports interactive changes in the position and properties of the springs and the dampers, the properties of the inner and outer drums, and the mass and position of the load. Simulation results are in good agreement with the experimental data. The virtual-lab has demonstrated to be a valuable design and analysis tool, allowing the user: (1) to get insight into the system behavior; (2) to tune the system parameters in order to improve the dynamic behavior; and (3) to simulate special events, such as a component breakage.

INTRODUCTION

Virtual-labs provide a flexible and user-friendly method to define the experiments performed on the model. Virtual-labs are composed of the *model* and the *view*. The *model* is the mathematical description of the system under study. The *view* is the user-to-model interface. It is intended to provide a visual representation of the simulated model behavior and to facilitate the user's interactive actions on the model during the simulation run. The graphical properties of the view elements are linked to the model variables, producing a bi-directional flow of information between the view and the model.

Modelica (Modelica 2007) is an object-oriented modeling language that facilitates the physical modeling paradigm. *MultiBody* (Otter et al. 2003) is a free Modelica library, providing 3-D mechanical components to model mechanical systems. An application of *MultiBody* library to the modeling of a household washing machine is described in (Donida et al. 2006).

An approach to the implementation of virtual-labs using Modelica language has been proposed in (Martin et al. 2006). This approach consists of:

- A systematic methodology to transform any Modelica model into a formulation suitable for interactive simulation has been proposed. Modelica models adapted according to this methodology can be used to set up interactive virtual-labs.
- *VirtualLabBuilder* Modelica library has been designed and programmed. It includes Modelica models implementing graphic interactive elements. These models allow the virtual-lab developer: (1) to compose the virtual-lab view; and (2) to link the visual properties of the view with the model variables. The interactive graphic interface is automatically generated during the model initialization process. The components of the library contain the code required to perform the communication between the view and the model.

Examples of virtual-lab implementation using this approach can be found in (Martin et al. 2007a, Martin et al. 2007b).

VIRTUAL-LAB IMPLEMENTATION USING *VirtualLabBuilder* MODELICA LIBRARY

The steps to define the virtual-lab using *VirtualLabBuilder* Modelica library are the following:

Step 1: Description of the Virtual-Lab Model. The Modelica model has to be adapted to suit interactive simulation following the methodology described in (Martin et al. 2006).

Step 2: Description of the Virtual-Lab View. *VirtualLabBuilder* library is composed of the packages shown in Figure 1a. Some of them are intended to be used by the virtual-lab developers (i.e., *VirtualLabBuilder* users):

- `ViewElements` and `VLabModels` packages contain the classes required to implement the virtual-lab view and to set up the complete virtual-lab. The `ViewElements` package includes interactive graphic elements that can be used to compose the virtual-lab view.
- `Examples` package contains some tutorial material illustrating the library use.

On the contrary, the classes within the `src` package are not intended to be directly used by the virtual-lab developers.

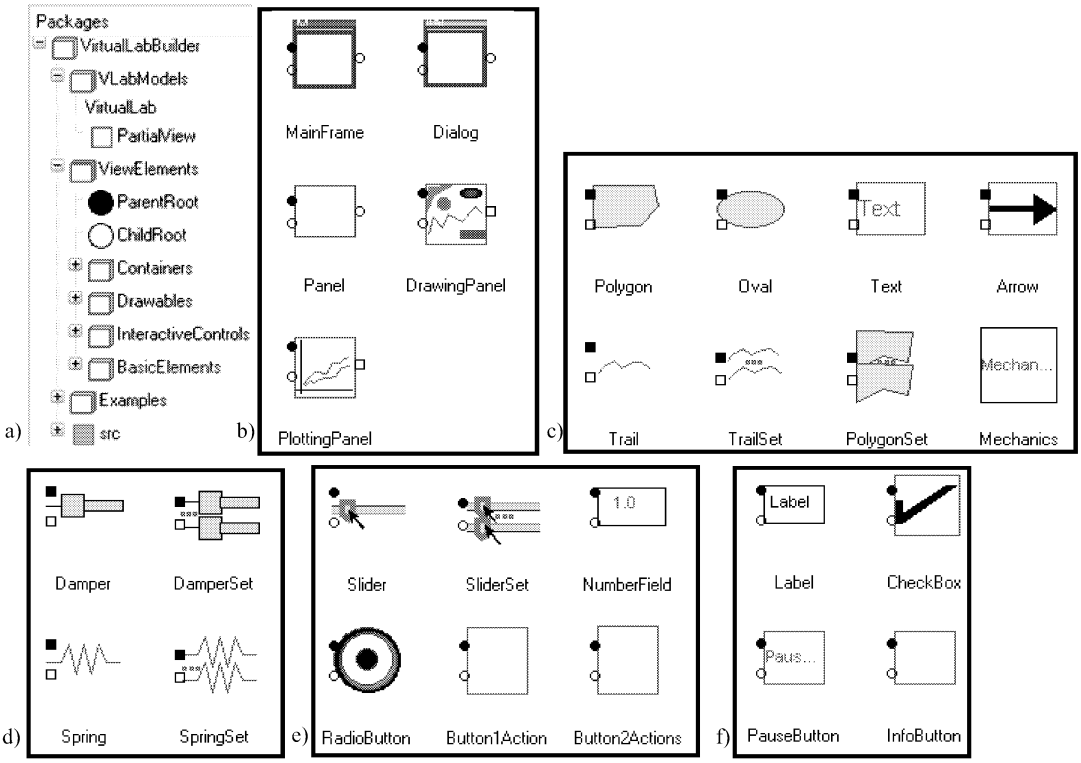


Figure 1: *VirtualLabBuilder* Library: a) General Structure; and Classes within the following Packages: b) `Containers`; c) `Drawables`; d) `Mechanics`; e) `InteractiveControls`; and f) `BasicElements`

The Modelica class describing the view must be a subclass of `PartialView` class. This class contains the code required to perform the communication between the model and the view. This code is valid for any model and view descriptions, and the virtual-lab designer only needs to set the length of the model-to-view communication interval.

The graphic components have to be connected by the virtual-lab developer. The "root" graphic component (i.e., the container component which hosts the rest of the components), named `root`, is pre-defined in the `PartialView` class. The connections among the graphic components determine their layout in the virtual-lab view. Modelica modeling environments – e.g. Dymola (Dynasim 2006) – allow defining in a drag-and-drop way the instantiation of the required *VirtualLabBuilder* library components and connecting them using the mouse.

Step 3: Virtual-Lab Set Up. The Modelica description of the virtual-lab has to be an instance of `VirtualLab` class. This class contains two parametrized generic classes: the class of the virtual-lab model and the class of the view. In addition, the virtual-lab designer has to specify, writing the required equations, how the variables of the model and the view classes are connected.

Step 4: Translation to Executable Code and Launch. The virtual-lab description is translated using Dymola and ran. As a part of the model initialization (i.e., the calculations performed to find the initial value of the model variables), the initial sections of the interactive graphic objects and of the `PartialView` class are executed. These initial sections contain calls to Modelica functions, which encapsulate calls to external C-functions that are Java-code generators.

As a result, during the model initialization, the Java code of the virtual-lab view is automatically generated, compiled and packed into a single jar file. Also, the communication procedure between the model and the view is set up. This communication is based on client-server architecture: the C-program generated by Dymola is the server and the Java program (which is automatically generated during the model initialization) is the client. Also, the communication procedure between the model and the view is set up. This communication is based on client-server architecture: the C-program generated by Dymola is the server and the Java program (which is automatically generated during the model initialization) is the client. Once the jar file has been created, it is executed. As a result, the initial layout of the virtual-lab view is displayed and the client-server communication is established. Then, the model simulation starts.

During the simulation run, there is a bi-directional flow of information between the model and the view. The model simulation (i.e., the server) sends to the view (i.e., the client) the data required to refresh the view. And the view sends to the model simulation the new value of the variables modified due to the user's interactive action.

Interactive graphic elements

There are four packages within the `ViewElements` package (see Figures 1a), which contain the graphic elements that can be used to define the virtual-lab view:

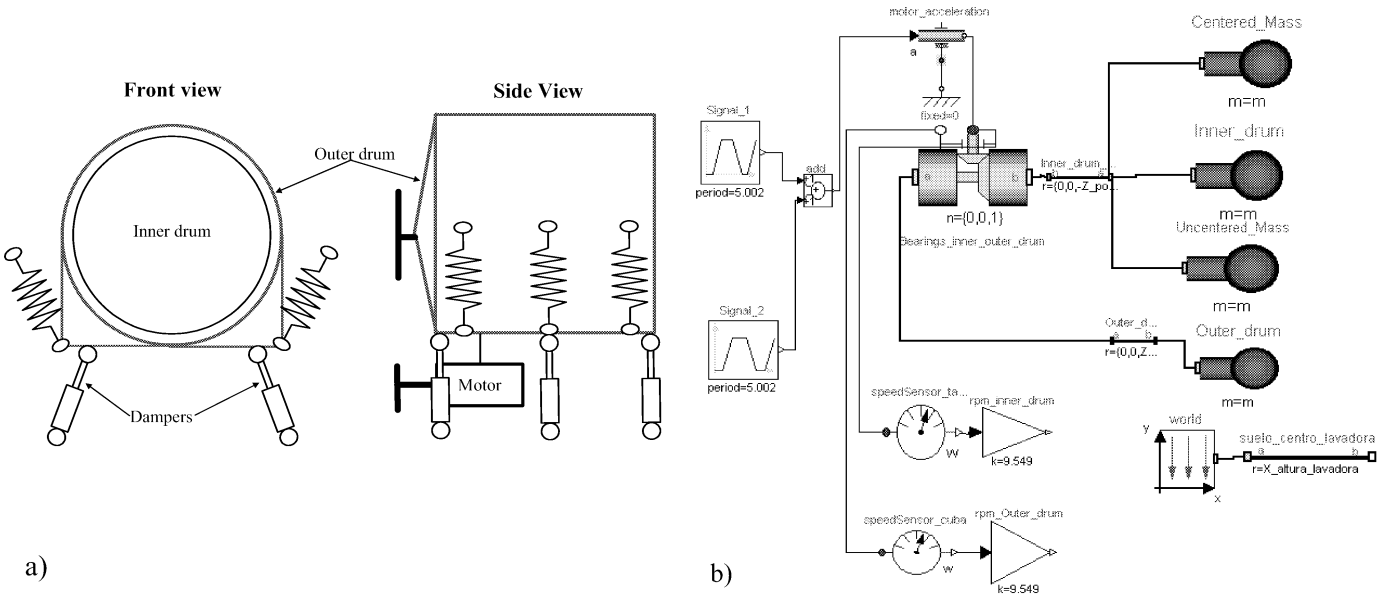


Figure 2: Drum-Type Washing Machine: a) Schematic Representation; and b) Diagram of the Modelica Dynamic Model

1. **`Containers`** package includes the following classes: `MainFrame`, `Dialog`, `Panel`, `DrawingPanel` and `PlottingPanel` (see Figure 1b). These graphic elements can host other graphic elements.

2. **`Drawables`** package includes the following classes: `Polygon`, `PolygonSet`, `Oval`, `Text`, `Arrow`, `Trail` and `TrailSet` (see Figure 1c); and also the `Mechanics` package (see Figure 1d), which contains the `Damper`, `DamperSet`, `Spring` and `SpringSet` classes. The elements of these classes must be placed inside containers, which provide the coordinate system. They are intended to be used for building an animated schematic representation of the system. The variables setting the geometric properties of these elements (position, size, etc.) can be linked to model variables.

3. **`InteractiveControls`** package includes six classes: `Slider`, `SliderSet`, `NumberField`, `RadioButton`, `Button1Action` and `Button2Actions` (see Figure 1e). Model variables can be linked to the variables defining the state of these interactive control elements. This allows changing the value of these model variables during the simulation run.

4. **`BasicElements`** package contains four classes: `Label`, `CheckBox`, `PauseButton` and `InfoButton` (see Figure 1f). Objects of these classes can be included inside a window or a panel. `PauseButton` class creates a button for pausing and resuming the simulation. `InfoButton` class creates a button for showing and hiding windows containing documentation in HTML format.

`Drawable` and `InteractiveControl` elements implement the information flow between the model and the view of the virtual-lab. The simulated value of the model variables modifies the properties of the drawable elements (i.e., model-to-view information flow). The user's interactive action on the interactive controls modifies the value of the model variables (i.e., view-to-model information flow).

WASHING MACHINE VIRTUAL-LAB

Drum-type washing machines are widely used in Europe. They are composed of an inner drum that rotates inside an outer drum, with a horizontal axis, making the clothes tumble upward and downward during washing cycle (see Figure 2a). During the drying cycle, clothes are submitted to both the gravity force (g) and the centrifugal force, generated by the inner drum rotational speed.

When the centrifugal force is bigger than "g", the clothes tend to stick to the inner drum wall. In some cases, it results in a non homogeneous distribution of the clothes' mass around the periphery of the inner drum. This is mainly due to the different composition of the tissues. Imbalance occurs when clothes' centre of mass does not coincide with the inner drum rotation axis, and it induces vibration to the outer drum.

In order to reduce the vibrations transmitted to the floor, the outer drum is suspended with springs. The forces transmitted to the frame (floor) can be drastically reduced if the resulting natural frequency (spring-drum) is very low. On the other hand, suspended drum movements can become uncontrollable when passing through the natural frequency and at low rotational speeds, which can cause collisions against the frame. Friction dampers are normally used to limit these movements.

Suspended drum movement depends on many factors. For example, the suspended mass inertia, the spring and damper positions and characteristics, the unbalanced mass value and location, and the spinning speed profile. All these parameters must be tuned for each new design, in order to minimize the drum displacements and the forces transmitted to the frame.

Accurate models of the drum dynamic, including unbalance load effects, can not be derived analytically due to the complexity of the dynamic behavior, influenced of all these parameters and their interactions. This limitation is even

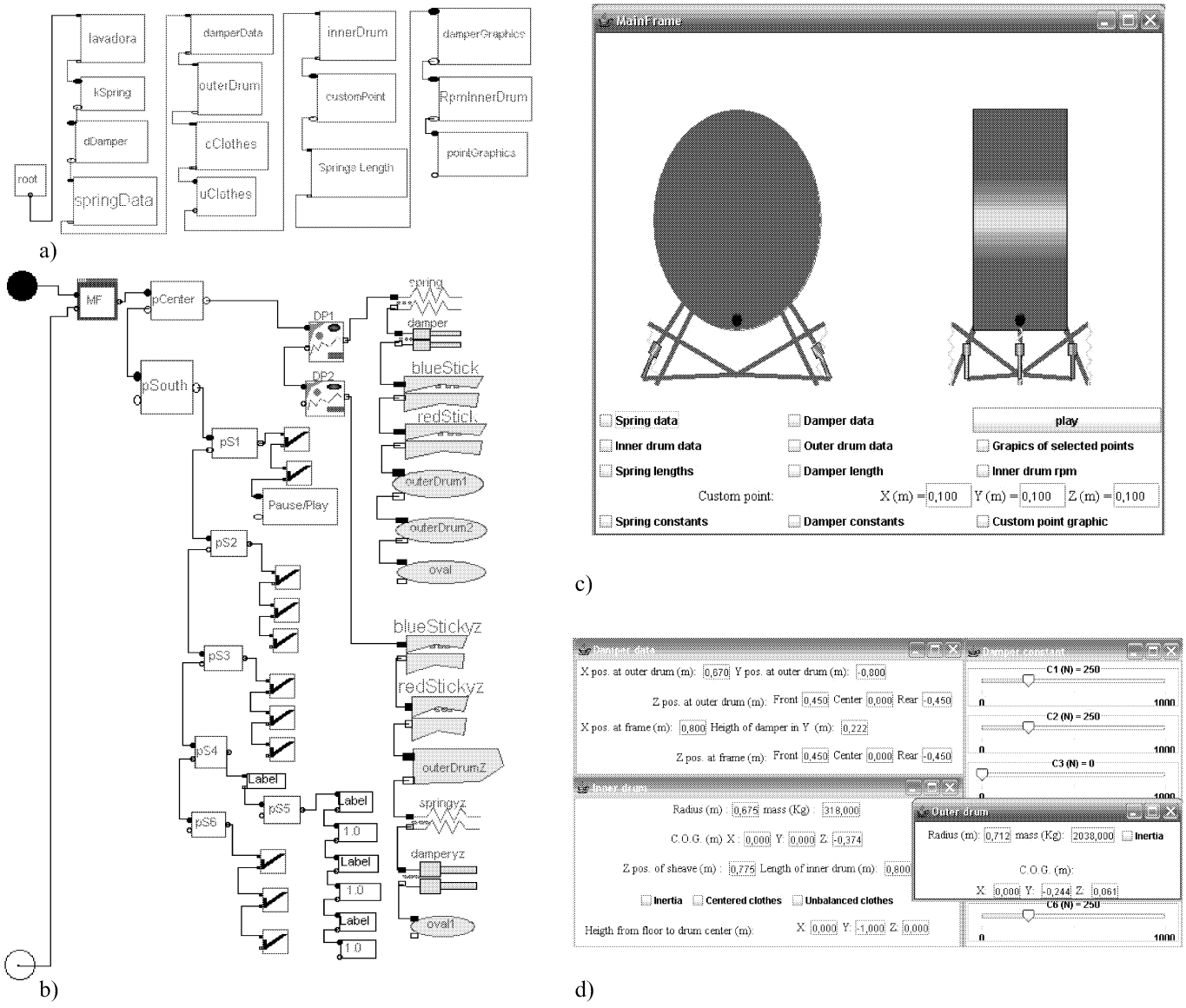


Figure 3: Washing Machine Virtual-Lab: a) Diagram of the View Description in Modelica; b) Diagram of the Main Window Description in Modelica; c) Main Window; and d) Some Dialog Windows

more evident when analyzing big-size washing machines (40 to 120 kg load capacity), which are suspended by several couples of springs and dampers. The dynamic behavior of the suspended drum can be successfully analyzed using rigid-body dynamic modeling and computer simulation.

Multibody model

MultiBody Modelica library (Otter et al. 2003) has been used for developing the dynamic model of the washing machine. All the bodies, except the springs, have been considered rigid. The diagram of the model is shown in Figure 2b. The suspended drum is composed of the following four bodies: outer drum, inner drum, centered and un-centered mass. These masses are attached to the inner drum. One rotational DoF is allowed between the inner and the outer drums.

The suspended drum has six DoF. Its dynamic behavior is governed by the forces generated by the mass of the un-centered clothes, the gravity, and the forces exerted by three pairs of springs and dampers. The springs and dampers are modeled as ideal elements (i.e., the force is proportional to the relative displacement or speed, respectively). Additionally, an external mass-free frame is considered. The springs and dampers are attached to this frame. This approach allows the computation of the floor reaction forces.

The model has been adapted – as it was proposed in (Martin et al. 2006) – to allow interactive changes in the position and properties of the springs and the dampers, the properties of the inner and outer drums, and the mass and position of the load. The model is intended to be used for tuning the value of these parameters, in order to improve the washing machine dynamic behavior. The evaluation of the suspended system displacement is accomplished for the following two critical test conditions: spinning start up and spinning at maximum speed. The dynamic behavior analysis is based on the following two key magnitudes: (1) the displacement of the suspended system with respect to the external frame; and (2) the forces transmitted to the floor. These forces can cause vibrations and relative displacements of the frame.

Virtual-lab view

The Modelica description of the virtual-lab view has been developed modularly, extending and connecting the required graphic components of the *VirtualLabBuilder* library (see Figure 3a). The virtual-lab view contains a main window and 15 dialog windows. Each of these windows is described by a class. The classes defining the main and the dialog windows are briefly described below.

The diagram of the Modelica class that defines the main window is shown in Figure 3b. The `MF` component – of `MainFrame` class – generates the window shown in Figure 3c. The `MF` layout policy is set to *BorderLayout*, in order to allow selecting the position of the hosted elements (i.e., north, south, center, east or west positions). It has two components hosted inside it, `pCenter` and `pSouth`, both of `Panel` class.

- `pCenter` component is placed in the center of `MF`, and it contains two containers, `DP1` and `DP2`, of `DrawingPanel` class These two components contain the frontal and the lateral animated diagrams of the washing machine. These diagrams are composed of several drawable elements of `Polygon`, `PolygonSet`, `Oval`, `DamperSet` and `SpringSet` classes.
- `pSouth` component hosts several interactive controls of `PauseButton`, `CheckBox`, `Label` and `NumberField` classes. The checkboxes allow to show and to hide the dialog windows. The button allows to pause and to resume the simulation. The spatial coordinates of system points are set using `NumberField` class components.

There are two types of dialog windows: (1) the windows containing plots that display the time evolution of some model variables; and (2) the windows containing interactive controls that allow the user to perform interactive changes in the model variables. The following windows contain the interactive controls (see Figure 3d):

- "*Spring data*"/"*Damper data*" windows allow changing the position of the spring/damper extremities in relation to the frame and the outer drum.
- "*Inner drum*" window allows changing the value of the relevant properties of the inner drum, including radius, mass, length, center of gravity (C.O.G) position, center position and sheave position. Additionally, this window contains checkboxes that allow to show and to hide three dialog windows. These three dialog windows contain interactive control elements that allow changing the C.O.G. mass and position of the centered and unbalanced load, and the inertia matrix of the inner drum.
- "*Outer drum*" window allows changing the outer drum properties (i.e., radius, mass, inertia and C.O.G. position).
- "*Spring constant*"/"*Damper constant*" windows allow changing the value of the spring/damper constants.

The virtual-lab contains five plot windows displaying the time-evolution of the following magnitudes: (1) the damper lengths; (2) the spring lengths; (3) the position of a system point, which can be interactively chosen by the virtual-lab user; (4) the position of certain relevant points of the system; and (5) the rotational speed of the inner drum.

Virtual-lab set up

The virtual-lab description is obtained as discussed in (Martin et al. 2006). It is translated using Dymola and executed. Then, the jar file containing the Java code of the virtual-lab view is automatically generated and executed, and the virtual-lab view is displayed. Simulation results are in good agreement with measurements of the suspended system displacements.

CONCLUSIONS

MultiBody and *VirtualLabBuilder* Modelica libraries have been used to implement the virtual-lab of a drum-type washing machine. The virtual-lab has been successfully applied for testing alternative designs of an industrial washing machine (120 Kg load capacity) manufactured by Fagor Industrial. This virtual-lab allows: (1) faster parameter tuning; and (2) simulating special events – e.g. spring or damper breakage. It constitutes a very valuable tool for the designer, that can online explore "what happens if" with no need to run again the simulation, enhancing his knowledge of the system behavior.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish CICYT, under DPI2004-01804 grant, and by the IV PRICIT (Plan Regional de Ciencia y Tecnología de la Comunidad de Madrid, 2005-2008), under S-0505/DPI/0391 grant.

REFERENCES

Donida F., G. Ferreti and F. Schiavo. 2006. "Modelling and Simulation of a Washing Machine", *In Proc. of the 50th Int. Congress ANIPLA 2006*, Rome (Italy).

Dynasim. 2006. "Dymola. User's Manual". Dynasim AB. Lund, Sweden.

Martin C., A. Urquia and S. Dormido. 2006. "An Approach to Virtual-lab Implementation using Modelica". *In Proc. of the 20th Annual European Simulation and Modelling Conference*, 137-141, Toulouse (France).

Martin C., A. Urquia and S. Dormido. 2007a. "Virtual-lab of a Solar House implemented using VirtualLabBuilder Modelica library". *In Proc. of the Conference on Systems and Control*, Marrakech (Morocco).

Martin C., A. Urquia and S. Dormido. 2007b. "Implementation of Interactive Virtual laboratories for Control Education using Modelica". *In Proc. of the European Control Conference,* Kos (Greece).

Modelica. 2007. http://www.modelica.org/

Otter M, H. Elmqvist and S.E. Mattsson. 2003. "The New Modelica MultiBody Library". *In Proc. of the 3rd Int. Modelica Conference,* 310-330, Linköping (Sweden).

PETRI NETS SIMULATION

ALGEBRAIC TECHNIQUES & SYMMETRIES: AN EFFICIENT SIMULATION APPROACH FOR WELL-FORMED NETS

Lorenzo Capra
Università degli Studi di Milano
Dipartimento di Informatica e Comunicazione
Via Comelico 39, 20131, Milano (Italy)
email: capra@dico.unimi.it

KEYWORDS
CPN, symbolic enabling test, symbolic marking, simulation.

ABSTRACT

Determining the set of transition color instances enabled in a given marking of a Colored Petri Net (the most famous High-level PN paradigm) is a basic task affecting all the analysis techniques based on state-space exploration. A symbolic approach to enabling test has been recently proposed in the context of Well-formed Nets (WN), a known CPN flavor retaining expressive power. The approach relies upon direct manipulation of WN arc functions by means of algebraic rewriting rules. The most interesting feature of the WN formalism lies in the ability of exploiting behavioral system's symmetries, thanks to structured color annotations. This paper semi-formally illustrates how the symmetry-based techniques typical of WN can be integrated to the WN algebraic enabling test, giving rise to a fully symbolic simulation kernel. Some experimental evidences of resulting efficiency improvements are presented.

INTRODUCTION

Well-formed Nets (WN) [Chiola et al., 1997] are a power-equivalent Colored Petri Net (CPN) [Jensen, 1997] subclass, suited to model and analyze discrete-event systems exhibiting behavioral symmetries. Stochastic WNs (SWN) [Chiola et al., 1993] offers the possibility of studying performances of systems through the underlying stochastic process. Effective state-space exploration algorithms relying upon the key concept of symbolic marking are peculiar to (S)WNs.

The traditional enabling test for colored transitions was recognized to be a critical task affecting discrete-event simulation [Gaeta, 1996], but it also plays a crucial role in state-space generation and model checking. An algebraic approach to enabling test in WN has been introduced [Capra and De Pierro, 2006], using the structural calculus (for WN) presented in [Capra et al., 2005].

The approach proposed in [Capra and De Pierro, 2006] moves the task complexity from run-time to algebraic rewriting of WN arc functions, with the aim of finding a formal expression for the "enabling function", that takes a marking as argument, and exactly results in the set of enabled transition instances. The algorithm operates symbolically, i.e., without any unfolding of WN nodes.

The WN symbolic enabling test [Capra and De Pierro, 2006] works on ordinary markings. This paper provides a core theoretical background for extending the technique to symbolic markings (SM), in order to exploit symmetries encoded into WNs. The algorithm works directly on SMs by manipulation of symbolic color tuples, and benefits from the minimality intrinsic to the SM canonical representation. Experimental evidences of achieved time/memory efficiency improvements on a complex WN model are also presented.

A semi-formal presentation style is adopted. First the WN syntax is recalled, along with the SM concept. Then the WN symbolic approach to enabling test is outlined. The following section illustrates how that approach can be efficiently merged to the symbolic firing mechanism typical of WN. Some experimental evidences are finally reported showing the benefits of the merging. An outline of the work and a discussion about possible evolutions conclude the paper.

WN BASICS AND SM NOTION

This section introduces the basic syntax of WNs through a simple example (Figure 1). We omit features typical of Stochastic WN such as priorities and rates, and we assume that the reader has some familiarity with basic PN notions.

As usual in Colored Petri Nets, places ($p1, p2$) may con-

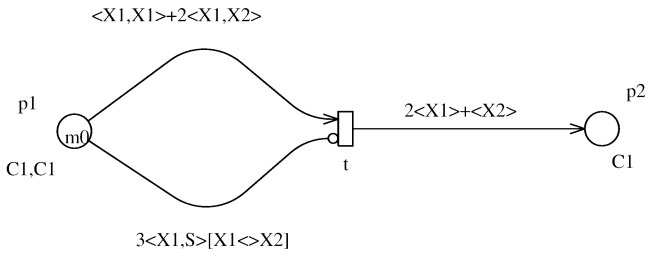


Figure 1: A Simple WN Model.

tain "colored" tokens of different identities. Transitions (t) are colored too, so that one speaks of "color" instances of a transition. In other words each WN node e is associated to a *color domain*, $\mathcal{C}(e)$, defined as Cartesian product of fi-

nite *basic color classes*, that are denoted by symbols C_i (in the example there is only one class, C_1). Tokens staying at place $p1$, as well as color instances of transition t, are pairs $\langle c_1, c_2\rangle \in C_1 \times C_1$. A basic color class may be partitioned into *static subclasses* $C_{i,j}$.

A marking $\mathbf{m}$ maps each place p to an element in $Bag(\mathcal{C}(p))$, $Bag(A)$ denoting the set of multisets defined over A. $\mathbf{m_0}$ denotes the initial marking.

Input, output and inhibitor arcs are labeled by functions $\mathcal{C}(t) \rightarrow Bag(\mathcal{C}(p))$, denoted $W^-(t,p)$, $W^+(t,p)$ and $H(t,p)$, respectively. An arc function is a linear combination of *function-tuple(s)*, $F : \langle f_1, \ldots, f_n\rangle$. Each f_i is a function $\mathcal{C}(t) \rightarrow Bag(C_j)$, C_j occurring on i-th position in $\mathcal{C}(p)$. Letting $t_c : \langle c_1, \ldots, c_m\rangle \in \mathcal{C}(t)$, then $F(t_c) = f_1(t_c) \times \ldots f_n(t_c)$, where $\times$ denotes the multiset Cartesian product. f_i is expressed in terms of *elementary* functions: the basic one is the *projection* X_k ($k \leq m$), defined as $X_k(t_c) = c_k$. A successor operator ! can be applied to X_k when C_j elements are *circularly* ordered. S and $S_{j,k}$ map any t_c to $\sum_{c \in C_j} 1 \cdot c$ and $\sum_{c \in C_{j,k}} 1 \cdot c$, respectively. Finally, $(S - X_k)(t_c)$ results in the complement of c_k.

The function application (Figure 1) $W^-(t,p1)(\langle c_1, c_2\rangle)$ results in $1 \cdot \langle c_1, c_1\rangle + 2 \cdot \langle c_1, c_2\rangle$.

A guard $[g]$ is associated to each transition t ($[true]$ is omitted), acting as filter for t's color instances. Guards are built upon basic predicates testing equality between projection applications, and membership to a given static subclass. For instance, $[X_1 \neq X_2](\langle c_1, c_1\rangle) = \emptyset$. Guards may be composed to function-tuples (see e.g. $H(t,p1)$): $F[g](t_c) = F(t_c)$ if $[g](t_c) = t_c$, otherwise $= \emptyset$.

$t_c \in \mathcal{C}(t)$ *is enabled* in $\mathbf{m}$ iff (i) $W^-(t,p)(t_c) \leq \mathbf{m}(p)$, (ii) $H(t,p)(t_c) > \mathbf{m}(p)$, for each place p, and (iii) $[g](t_c) = t_c$ ($>, \leq, +, -$ are implicitly extended to multisets). It can fire, leading to $\mathbf{m}'$: $\forall p$

$\mathbf{m}'(p) = \mathbf{m}(p) + W^+(t,p)(t_c) - W^-(t,p)(t_c)$.

The peculiar feature of WN formalism is the ability of capturing behavioral symmetries thanks to a structured syntax. Through the notion of Symbolic Marking (SM), it is possible to build an aggregated state space (Symbolic Reachability Graph, or SRG [Chiola et al., 1997]) or to perform a symbolic discrete-event simulation.

Symbolic Marking

A SM provides a syntactical equivalence relation on ordinary markings: two markings belong to the same SM iff they can be obtained from one another by means of *permutations* on unordered color classes that preserve static subclasses, and *rotations* on ordered color classes. The definition of a SM $\widehat{\mathbf{m}}$ comprises two parts: one specifying the so called *dynamic subclasses* and another representing the distribution of symbolic tokens over places.

Dynamic subclasses define a *parametric partition* of basic color classes that preserves static subclasses. $\widehat{C}_i$ denotes the set of C_i dynamic subclass (in $\widehat{\mathbf{m}}$). Let s_i be the number of C_i static subclasses. $Z_j^i \in \widehat{C}_i$ refers to a static subclass, denoted $d(Z_j^i)$, $1 \leq d(Z_j^i) \leq s_i$, and has an associated cardinality.[1] It holds $\forall k : 1 \ldots s_i$, $|C_{i,k}| = \sum_{j:d(Z_j^i)=k} |Z_j^i|$.

If C_i is an ordered class (in that case $s_i = 1$), then its dynamic subclasses must be ordered accordingly, formally there is a homomorphism $h : \widehat{C}_i \rightarrow \widehat{C}_i$ such that $h^{|\widehat{C}_i|} = h$. Intuitively, contiguous dynamic subclasses represent contiguous (sets of) colors.

$\widehat{\mathbf{m}}$ maps each place p to an element in $Bag(\widehat{\mathcal{C}}(p))$, $\widehat{\mathcal{C}}(p)$ (the *symbolic color domain* of p) being obtained by replacing each C_i in $\mathcal{C}(p)$ by $\widehat{C}_i$.

Among several possible representations for a SM, two are of particular relevance. The *canonical representation* [Chiola et al., 1993], that minimizes dynamic subclasses and provides an univocal representation (unless otherwise specified we will implicitly refer to it). On the opposite there is the *split representation*, with all dynamic subclasses of cardinality one. The split representation of $\widehat{\mathbf{m}}$ is obtained replacing each Z_j^i, $|Z_j^i| > 1$, by $\sum_{k=1}^{|Z_j^i|} Z_{j,k}^i$ (double subscripts denote dynamic subclasses deriving from splitting). Splitting must preserve cardinality, static subclasses, and ordering.

Let us consider the net on Figure 1: an example of SM that will be used also in the sequel is $\widehat{\mathbf{m}}_1$ (place $p2$ is assumed empty):

$$\widehat{\mathbf{m}}_1(p1) : \; 2 \cdot \langle Z_1, Z_1\rangle + 3 \cdot \langle Z_2, Z_1\rangle$$
$$\widehat{C}_1 = \{Z_j\}, \; |Z_1| = 2, \; |Z_2| = 1$$

Let $C_1 = \{c_1, c_2, c_3, \ldots\}$. An ordinary marking $\mathbf{m}'$ belonging to $\widehat{\mathbf{m}}_1$ is

$$2\langle c_1, c_1\rangle + 2\langle c_2, c_2\rangle + 2\langle c_1, c_2\rangle + 2\langle c_2, c_1\rangle + 3\langle c_3, c_1\rangle + 3\langle c_3, c_2\rangle.$$

The split representation of $\widehat{\mathbf{m}}_1$ ($split(\widehat{\mathbf{m}}_1)$) is:

$$2\langle Z_{1,1}, Z_{1,1}\rangle + 2\langle Z_{1,2}, Z_{1,2}\rangle + 2\langle Z_{1,1}, Z_{1,2}\rangle + 2\langle Z_{1,2}, Z_{1,1}\rangle + 3\langle Z_2, Z_{1,1}\rangle + 3\langle Z_2, Z_{1,2}\rangle$$

Let $\mathcal{C}(t)$ be the color domain of transition t and $\widehat{\mathbf{m}}$ be a *split* SM. The *symbolic color domain* of t (respective to $\widehat{\mathbf{m}}$), denoted $\widehat{\mathcal{C}}(t)$, is obtained replacing each C_i in $\mathcal{C}(t)$ by $\widehat{C}_i$. A symbolic color instance is a tuple $\widehat{c} \in \widehat{\mathcal{C}}(t)$. Some symbolic color instances of t respective to $split(\widehat{\mathbf{m}}_1)$ are: $\langle Z_{1,1}, Z_{1,2}\rangle$, $\langle Z_{1,2}, Z_{1,1}\rangle$, $\langle Z_2, Z_{1,1}\rangle$.

The semantics of arc functions is unchanged when considering symbolic color domains, by the way their arity becomes $\widehat{\mathcal{C}}(t) \rightarrow Bag(\widehat{\mathcal{C}}(p))$. As an example (Figure 1, inhibitor arc function) the application $H(t,p1)(\langle Z_{1,1}, Z_{1,2}\rangle)$ results in $\sum_{z \in \widehat{C}_1} 3 \cdot \langle Z_{1,1}, z\rangle$.

As long as we consider split SMs, the enabling notion is straightforwardly extended to symbolic color instances: the enabling of $\widehat{c} : \langle Z_{1,1}, Z_{1,2}\rangle$ in $\widehat{\mathbf{m}_1}$ is proved by evaluating $W^-(t,p1)(\widehat{c}) \leq split(\widehat{\mathbf{m}_1})(p) < H(t,p1)(\widehat{c})$. Note that $\langle Z_{1,1}, Z_{1,2}\rangle \equiv \langle Z_{1,2}, Z_{1,1}\rangle$ up to a permutation on split dynamic subclasses, i.e., these instances fold together the same ordinary instances.

[1] superscripts i are omitted when there is only one color class.

Testing enabled symbolic instances in the traditional way is highly inefficient for two major reasons: splitting of $\widehat{\mathbf{m}}$ is forced, as a consequence, possible equivalences are not taken into account. Our idea is to extend the symbolic enabling test for WN [Capra and De Pierro, 2006] to symbolic markings, avoiding splitting unless strictly necessary. The combination of two symbolic approaches should yield a very efficient algorithm.

SYMBOLIC ENABLING TEST OVERVIEW

In opposition to a brute-force application of the enabling expression (for each place and transition color instance), the problem of determining enabled transition instances in a given marking may be stated in terms of looking for a function, $EN(\mathbf{m}, t)$, which given $\mathbf{m}$ and t, returns t's enabled instances. It has been recently approached in [Capra and De Pierro, 2006], with the idea of deeply exploiting WN's structured syntax. With respect to similar approaches [Gaeta, 1996, Ilie and Rojas, 1993], that are heuristics-based and may be applied only when WN arc functions are matching restrictive patterns, the technique proposed in [Capra and De Pierro, 2006] deals with all WN features and operates in a fully symbolic way, i.e., without any unfolding of WN nodes. It operates by rewriting arc functions W^- and H by means of operators acting on function-tuples (symbol $\longrightarrow$ will denote a rewriting application), exploiting the algebraic properties of a structural calculus for WN [Capra et al., 2005].

The algebraic WN framework is characterized by a high-level language $\mathcal{L}$, closed with regard to a core-set of functional operators. Terms in $\mathcal{L}$ are a simple extension of WN arc functions: guards may be left-composed to function-tuples (in that case they are called filters), tuple components may be *intersections* of elementary functions, last the k-th successor ($!^k$) is introduced, so $3 \cdot [X_1 \neq X_2]\langle !^2 X_1 \cap (S - X_2), X_1\rangle$ belongs to $\mathcal{L}$. We can assume without losing generality that any $E_j \in \mathcal{L}$ is a linear combination of function-tuples mapping on sets.

Functional Operators The operators mainly involved in the computation of $EN(\mathbf{m}, t)$ are intersection and transpose. Let $F, F' : A \rightarrow Bag(B)$.

Transpose $F^t : B \rightarrow Bag(A)$ is defined as $F^t(b)(a) = F(a)(b)\ \forall a \in A, \forall b \in B$ ($m(c)$ denotes the multiplicity of element c on multi-set m). Given $c \in \mathcal{C}(p)$, $W^-(t,p)^t(c)$ maps to t's instances which need for a number of c on place p in order for being enabled, along with the quantity.

Intersection $(F \cap F')(a)$ is defined as $F(a) \cap F'(a)$ (letting $m, m' \in Bag(C)$, then $(m \cap m')(c) = \min(m(c), m'(c))$, $\forall c \in C$). Functions F, F' are said disjoint if $F \cap F'$ is equal to the empty (multi)set constant $\emptyset$.

A key property of $\mathcal{L}$ is that any language expression can be rewritten (using the intersection rules) into a formal sum of disjoined terms. As an example:
$2\langle S, X_2\rangle + \langle X_1, X_2\rangle \longrightarrow 3\langle X_1, X_2\rangle + 2\langle S - X_1, X_2\rangle$

Enabling function

The definition of $EN(\mathbf{m}, t)$ is modular ($^\bullet t \cup {}^\circ t$ is the set of places connected to t via input/inhibitor arcs): $EN(\mathbf{m}, t) = \bigcap_{p \in {}^\bullet t \cup {}^\circ t} EN_{(t,p)}(\mathbf{m})$, where $EN_{(t,p)}(\mathbf{m})$ is the set of t color instances that are *locally* enabled in $\mathbf{m}$, regarding p. $EN_{(t,p)}$ is formally expressed in terms of $\mathcal{L}$.
All the information about local enabling of t are contained in the transposes of $W^-(t,p)$ and $H(t,p)$. One might build an enabling table which enumerates, for each $c'_i \in \mathcal{C}(t)$, $c \in \mathcal{C}(p)$, an interval of values representing the number of occurrences of c on p that ensure the local enabling of c'_i. By operating the framework's transpose and disjunction algorithms on functions $W^-(t,p)$ and $H(t,p1)$ we can build the so-called "symbolic local enabling table" (Table 1) for the running example. This table gives a compact representation of local enabling tables, and may be equivalently expressed as a formal sum $EN_{(t,p1)}$ that provides a pattern for locally enabled instances of t in a generic marking, regarding place $p1$ (symbol $\oplus$ is used to avoid confusion with multiset sum):

$$\begin{array}{l}
\lfloor 1,2\rfloor\ \langle X_1, S - X_1\rangle[X_1 = X_2] \oplus \\
\lfloor 2,2\rfloor\ \langle X_1, X_2\rangle[X_1 \neq X_2] \oplus \\
\lfloor 0,2\rfloor\ \langle X_1, (S - X_1) \cap (S - X_2)\rangle[X_1 \neq X_2] \oplus \\
\lfloor 3,\infty\rfloor\ \langle X_1, X_1\rangle[X_1 = X_2]
\end{array}$$

Each term T_k of $EN_{(t,p1)}$ is associated to an interval $[\alpha_k, \beta_k]$, defining enabling bounds. The fourth term says for $\langle c_1, c_1\rangle$ to be enabled, it is necessary and sufficient that at least three tokens $\langle c_1, c_1\rangle$ occur on $p1$. The first three terms say necessary and sufficient condition for $\langle c_1, c_2\rangle$, $c_2 \neq c_1$, to be enabled is that : 1) one or two tokens $\langle c_1, c_1\rangle$ occur on $p1$, 2) two tokens $\langle c_1, c_2\rangle$ occur on $p1$, 3) at most two tokens $\langle c_1, c_3\rangle$, $c_3 \neq c_1 \wedge c_3 \neq c_2$, occur on $p1$.
The computation of $EN_{(t,p1)}(\mathbf{m})$ is accomplished only considering colors (tokens) occurring on $\mathbf{m}(p1)$, using information on the cardinality of $W^-(t,p1)$ (meant as cardinality of $W^-(t,p1)(c)$, assumed constant $\forall c$). The multiset $\mathcal{EN}$, defined as sum of terms $\alpha_k \cdot T_k(c')$ such that $\alpha_k > 0$ and the occurrences of c' on $p1$ range over $[\alpha_k, \beta_k]$, provides an expression for t's instances satisfying the (local) enabling condition due to the input arc: they match $\mathcal{EN}$ elements with multiplicity n, as the tokens required by any arc function application. Consider the ordinary marking $\mathbf{m}'$ defined in subsection on SM. In that case $\mathcal{EN}$ is:

$$\underbrace{1 \cdot \langle c_1, S - c_1\rangle}_{\langle c_1,c_1\rangle} + \underbrace{2 \cdot \langle c_1, c_2\rangle}_{\langle c_1,c_2\rangle} + \underbrace{1 \cdot \langle c_2, S - c_2\rangle}_{\langle c_2,c_2\rangle} + \underbrace{2 \cdot \langle c_2, c_1\rangle}_{\langle c_2,c_1\rangle}$$

The contributions due to the different colors occurring on $p1$ are outlined. Because of $|W^-(t,p1)| = 3$, potentially enabled color instances of t are obtained by intersecting $\mathcal{EN}$ terms whose multiplicity sum amounts 3:
$\langle c_1, S - c_1\rangle \cap \langle c_1, c_2\rangle \longrightarrow \langle c_1, c_2\rangle$;
$\langle c_2, S - c_2\rangle \cap \langle c_2, c_1\rangle \longrightarrow \langle c_2, c_1\rangle$;

Color instances actually enabled are finally obtained subtracting from the potential set just computed all terms $T_k(c'')$

Table 1: $(W^-)^t$ and H^t rewritten into reciprocally disjoined sums

$W^-(t,p1)^t$	$H(t,p1)^t$
$1 \cdot \langle X_1, S - X_1 \rangle [X_1 = X_2]$	$3 \cdot \langle X_1, S - X_1 \rangle [X_1 = X_2]$
$2 \cdot \langle X_1, X_2 \rangle [X_1 \neq X_2]$	$3 \cdot \langle X_1, X_2 \rangle [X_1 \neq X_2]$
$3 \cdot \langle X_1, X_1 \rangle [X_1 = X_2]$	$-$
$-$	$3 \cdot \langle X_1, (S - X_1) \cap (S - X_2) \rangle [X_1 \neq X_2]$

such that $\alpha_k = 0$, and the multiplicity of c'' on $p1$ exceeds β_k. In the example the minuend is: $\langle c_3, (S - c_3) \cap (S - c_1) \rangle \cup \langle c_3, (S - c_3) \cap (S - c_2) \rangle$, thereby

$$EN_{(t,p1)}(\mathbf{m}') = \{\langle c_1, c_2 \rangle, \langle c_2, c_1 \rangle\}.$$

EXTENSION TO SYMBOLIC MARKINGS

The symbolic local enabling table can be used to efficiently compute enabled symbolic color instances in a split SM. The only difference is that it applies to tuples of dynamic subclasses instead of basic colors.
As in the ordinary case, a) only symbolic colors occurring on $\widehat{\mathbf{m}}(p)$ must be considered; b) the symbolic multiset $\widehat{\mathcal{EN}}$ has to be defined by evaluating function-tuples forming $EN_{(t,p)}$ on symbolic colors. What may be done without any difficulty as long as we are considering split SMs.
All we need are intersection rewriting rules between symbolic tuples, listed (not exhaustively) on APPENDIX-Table 4. Basic set-theory rules (idempotence, De Morgan, etc.) are omitted. It is remarkable that each rule applies to dynamic subclasses of arbitrary cardinality, so it will work also on *canonical* SM. Intersection rules are applied with priority over whatever else rewriting rules.
During the calculus compact symbols such as S, $S - Z_j^i$, etc. are manipulated. At the end they should be expanded in order to obtain enabled symbolic color instances. Expansion rules are listed on the lower part of Table 4.
To highlight how the algorithm operates in the general case (but it would be more correct saying "worst" case), let us consider the running example's SM $\widehat{\mathbf{m}}_1$ (subsection on SM). Applying $EN_{(t,p1)}$ to $split(\widehat{\mathbf{m}}_1)$ results in the multiset $\widehat{\mathcal{EN}} \in Bag(\widehat{C}_1 \times \widehat{C}_1)$

$$\overbrace{1 \cdot \langle Z_{1,1}, S - Z_{1,1} \rangle}^{\langle Z_{1,1}, Z_{1,1} \rangle} + \overbrace{1 \cdot \langle Z_{1,2}, S - Z_{1,2} \rangle}^{\langle Z_{1,2}, Z_{1,2} \rangle} + \underbrace{2 \cdot \langle Z_{1,1}, Z_{1,2} \rangle}_{\langle Z_{1,1}, Z_{1,2} \rangle} + \underbrace{2 \cdot \langle Z_{1,2}, Z_{1,1} \rangle}_{\langle Z_{1,2}, Z_{1,1} \rangle}$$

Each term (symbolic tuple) of $\widehat{\mathcal{EN}}$ represents a set of symbolic color instances of t, and it comes from evaluating a corresponding term of $EN_{(t,p1)}$ on a (symbolic) color occurring on $p1$, as made for the ordinary case. Locally, potentially enabled symbolic color instances of t correspond to multiset elements of multiplicity 3, that are obtained by intersecting (non disjoint) terms whose coefficient sum is 3. Using the rules defined on Table 4 we obtain:

$\langle Z_{1,1}, S - Z_{1,1} \rangle \cap \langle Z_{1,1}, Z_{1,2} \rangle \longrightarrow \langle Z_{1,1}, Z_{1,2} \rangle$,

$\langle Z_{1,2}, S - Z_{1,2} \rangle \cap \langle Z_{1,2}, Z_{1,1} \rangle \longrightarrow \langle Z_{1,2}, Z_{1,1} \rangle$,

They are actually enabled instances (no final expansion is required in this case), since the minuend turns out to be

$\langle Z_2, (S - Z_2) \cap (S - Z_{1,1}) \rangle \cup \langle Z_2, (S - Z_2) \cap (S - Z_{1,2}) \rangle$.

Note that $\langle Z_{1,1}, Z_{1,2} \rangle \equiv \langle Z_{1,2}, Z_{1,1} \rangle$ so either one of them should be processed (in case of a circular ordering on class C_1 that would be no more true).
We can generalize, stating the property below.

Prop. 1 *$EN_{(t,p)}(split(\widehat{\mathbf{m}}))$ is the set of symbolic instances of t locally enabled in SM $\widehat{\mathbf{m}}$, regarding p.*

Partial splitting of SM

Very often a complete splitting of a *canonical* SM $\widehat{\mathbf{m}}$ is not necessary to compute enabled symbolic color instances. Carrying out splitting only when necessary makes faster the enabling test and reduces the overhead due to processing of equivalent instances. We propose a (three-phase) clever evaluation of WN functions $[g']T[g] \in \mathfrak{L}$ on symbolic colors $\widehat{c}$, based on algebraic term rewriting that minimizes both dynamic subclass splitting and intersections. The evaluation of a possibly guarded function-tuple is treated first, then the (possible) application of a left-composed guard (filter) $[g']$ (filters can be generated by transpose operation), finally a clever expansion of resulting tuples is performed.
The linear extension of a WN function $F \in \mathfrak{L}$ will be used.

Def. 1 *Let $F : \widehat{A} \to Bag(\widehat{B})$, $\widehat{A}$ and $\widehat{B}$ being symbolic color domains defined over $\widehat{\mathbf{m}}$. Then*

$$F(\widehat{c}) = \sum_{\widehat{c}' \in split(\widehat{c})} F(\widehat{c}')$$

$split(\widehat{c})$ being the set of symbolic tuples of cardinality one dynamic subclasses corresponding to $\widehat{c}$.

Clever Evaluation of $T[g](\widehat{c})$ Consider the application of $F_2 : \langle X_1, X_2 \rangle [X_1 \neq X_2]$ (the 2nd term of Table 1) to $\langle Z_1, Z_1 \rangle$, $|Z_1| = 2$. Given that $split(\langle Z_1, Z_1 \rangle) = \{\langle Z_{1,1}, Z_{1,1} \rangle, \langle Z_{1,2}, Z_{1,2} \rangle, \langle Z_{1,1}, Z_{1,2} \rangle, \langle Z_{1,2}, Z_{1,1} \rangle\}$,
according to Definition 1:
$F_2(\langle Z_1, Z_1 \rangle) = \langle Z_{1,1}, Z_{1,2} \rangle + \langle Z_{1,2}, Z_{1,1} \rangle$

Splitting has been necessary to correctly evaluate the guard, what would be otherwise impossible due to the repetition of dynamic subclass Z_1 on the tuple-argument (the evaluation of guards on split symbolic tuple-arguments comes to be

straightforward). No splitting would be necessary instead to evaluate $F_2(\langle Z_1, Z_2\rangle)$ $(= \emptyset)$, since dynamic subclasses define a parametric *partition* of colors.
On table 5 are listed the cases in which guards can be evaluated (or simplified) without splitting dynamic subclasses taken as arguments. A tuple of dynamic subclasses is denoted $\langle ds_1, \ldots, ds_m\rangle$. Predicates are assumed in conjunctive form without loss of generality.
The next properties precisely set which dynamic subclasses have to be split in order to correctly evaluate a guarded function-tuple in $\mathcal{L}$. Let $\widehat{A}, \widehat{B}$ be symbolic color domains defined on a canonical SM. Let $\widehat{C}_{X_i} = \widehat{C}_j$, C_j being X_i's co-domain. Guard evaluation is considered first.

Prop. 2 *Let $[g] : \widehat{A} \to \widehat{A}$. Then $[g](\widehat{c})$ can be directly evaluated (resulting either in $\widehat{c}$ or ϕ) if, after applying the rewritings of Table 5, either the outcome is ϕ or the following hold:*

- *if a basic predicate $X_i = (\neq)X_j$ occurs on $[g]$ and $ds_i = ds_j$ then $|ds_i| = 1$*
- *if a successor symbol $!^k X_i$ occurs on $[g]$ then $\widehat{C}_{X_i}$ is split.*

As an example, assuming $|Z_j^1| = 1, \forall j$:
$[X_2 = !^2 X_1 \wedge X_3 \neq X_4]\langle Z_1^1, Z_3^1, Z_1^2, Z_2^2\rangle \longrightarrow$
$[X_2 = !^2 X_1]\langle Z_1^1, Z_3^1, Z_1^2, Z_2^2\rangle \longrightarrow \langle Z_1^1, Z_3^1, Z_1^2, Z_2^2\rangle$.[2]

When evaluating a function-tuple we have to take color multiplicities into account (we are considering the function linear extension).
Consider $F' : \langle S - X_1, S - X_1\rangle$ and the tuple-argument $\langle Z_1, Z_2\rangle$, $|Z_1| = 2$, $|Z_2| = 3$ ($|split(\langle Z_1, Z_2\rangle)| = 6$). Splitting is necessary to correctly evaluate the function-tuple, due to the repetition of projection symbol X_1. Only dynamic subclass Z_1 (the one "projected" by X_1) needs to be split, in fact the application $F'(\langle Z_1, Z_2\rangle)$ (Definition 1) results in $3 \cdot \langle S - Z_{1,1}, S - Z_{1,1}\rangle + 3 \cdot \langle S - Z_{1,2}, S - Z_{1,2}\rangle$. Efficient function-tuple evaluation is formalized next.

Prop. 3 *Let $T = \langle f_i, \ldots, f_n\rangle \in \mathcal{L}$, $T : \widehat{A} \to Bag(\widehat{B})$, $\widehat{c} = \langle ds_1, \ldots, ds_m\rangle \in \widehat{A}$. Let $DS = \{ds_j\}$ s.t. $\nexists X_j$ in T. If the following conditions hold:*

- *if projection symbol X_i (or its complement) is repeated on T then $|ds_i| = 1$*
- *if the successor symbol $!^k X_i$ occurs on T then $\widehat{C}_{X_i}$ is split.*

Then $T(\widehat{c}) = K \cdot \langle f_1(\widehat{c}), \ldots, f_n(\widehat{c})\rangle$, where

$X_j(\widehat{c}) = ds_j$, $!^k X_j(\widehat{c}) = !^k ds_j$, $(S - !^k X_j)(\widehat{c}) = S - !^k ds_j$,
$S(\widehat{c}) = S$, $S_{i,j}(\widehat{c}) = S_{i,j}$, $(f_{j_1} \cap f_{j_2})(\widehat{c}) = f_{j_1}(\widehat{c}) \cap f_{j_2}(\widehat{c})$,

$$(S - X_j)(\widehat{c}) = \begin{cases} S - ds_j & \text{if } |ds_j| = 1 \\ |ds_j| \cdot (S - ds_j) + & \\ (|ds_j| - 1) \cdot ds_j & \text{otherwise} \end{cases}$$

$$K = \begin{cases} \prod_{ds_j \in DS} |ds_j| & \text{if } DS \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

[2] by convention $!^k Z_j^i$ $(1 \leq k < |C_i|) = Z^i_{(j-1+k)_{mod|C_i|}+1}$.

Summarizing, for being able to correctly evaluate a guarded function-tuple $T[g](\widehat{c})$ we only need to split symbolic color classes $\widehat{C_i}$ such that there is some symbol $!^k X_j$ (of class-i) occurring on T or in, and dynamic subclasses ds_i occurring on $\widehat{c}$ that are either repeated or referred to by a projection symbol X_i repeated on T.
During the calculus intersections between multisets of symbolic tuples have to be carried out. Cor.1 (an obvious consequence of the multi-set intersection definition), and Cor.2 (a particular case where De Morgan applies to multisets, that we can always refer to) permit intersection between (symbolic) multisets to be solved using set-intersection rules (listed on Table 4).

Cor. 1 *Let $A \cup B \cup C \subseteq \widehat{D}$ ($\widehat{D}$ denoting a symbolic color domain), $\{k_1, k_2, k_3\} \subseteq \mathbb{N}$. Then*
$k_1 \cdot A \cap k_1 \cdot B = \min(k_1, k_2) \cdot (A \cap B)$.

Cor. 2 *Let $\{ma, mb, mc\} \subseteq Bag(\widehat{D})$, $ma \cap mb = \phi$. Then*
$(ma + mb) \cap mc = ma \cap mc + mb \cap mc$

Consider the following function-tuples:

$i)\ \langle X_1, S - X_2\rangle$
$ii)\ \langle (S - X_1) \cap (S - X_2), X_3\rangle$
$\widehat{C}_1 \times \widehat{C}_1 \to Bag(\widehat{C}_1 \times \widehat{C}_1 \times \widehat{C}_1)$

and the tuple-argument:

$\langle Z_1, Z_2, Z_3\rangle$, $|Z_1| = |Z_2| = 2$, $|Z_3| = 3$

According to Property 3 no dynamic subclasses are split, their evaluations result in (the final expansion meets the multiset Cartesian product)

$$\begin{aligned} i)\quad & 3 \cdot \langle Z_1, 2 \cdot (S - Z_2) + Z_2\rangle \\ \equiv\ & 6 \cdot \langle Z_1, S - Z_2\rangle + 3 \cdot \langle Z_1, Z_2\rangle \end{aligned}$$

$$\begin{aligned} ii)\quad & 1 \cdot \langle (2 \cdot (S - Z_1) + Z_1) \cap (2 \cdot (S - Z_2) + Z_2), Z_3\rangle \\ \equiv\ & 1 \cdot \langle 2 \cdot (S - Z_1) \cap (S - Z_2) + Z_2 + Z_1, Z_3\rangle \\ \equiv\ & 2 \cdot \langle (S - Z_1) \cap (S - Z_2), Z_3\rangle + \langle Z_1, Z_3\rangle + \langle Z_2, Z_3\rangle \end{aligned}$$

Clever Evaluation of $[g'](b)$ The evaluation of $T[g](\widehat{c})$ results in a weighted sum of symbolic tuples $\langle t_1, \ldots, t_n\rangle \subseteq \widehat{B}$, each tuple denoting a *set* of symbolic color instances. If the function-tuple T is preceded by a filter $[g']$ we need to evaluate $[g']$ on the multiset $b = T[g](\widehat{c})$ in order to produce the final result.
The notion of filter (guard) can be straightforwardly extended to multisets. Let $m \in Bag(A)$, $[g'](m) \in Bag(A)$ is defined as $\forall a \in A$: $[g'](m)(a) = m(a)$ if $g'(a) = true$, 0 otherwise. That means simplification rules on Table 5, Property 2, and the usual guard evaluation rules, can still apply to multisets of symbolic tuples.
An approach might consists of first splitting a tuple $\langle t_1, \ldots, t_n\rangle$, that might contain symbols such as $S, S - X_j$, etc., into its corresponding color instances (applying the expansion rules reported on Table 4), then evaluating the filter $[g']$ on them.

In many cases however it is more convenient to operate reductions on terms $[g'](\langle t_1, \ldots, t_n \rangle)$ before expansion: assume for instance that function i) just introduced were preceded by the filter $[X_1 = X_2]$. Applying the filter to function application's result:

$[X_1 = X_2](3 \cdot \langle Z_1, Z_2 \rangle) \stackrel{\text{Tab. 5}}{\longrightarrow} \phi$

$[X_1 = X_2](6 \cdot \langle Z_1, S - Z_2 \rangle) \longrightarrow \ldots$

The second term may either be expanded or (more conveniently) rewritten as

$\stackrel{\text{Tab. 6}}{\longrightarrow} 6 \cdot \langle (S - Z_2) \cap Z_1, (S - Z_2) \cap Z_1 \rangle \stackrel{\text{Tab. 4}}{\longrightarrow}$
$6 \cdot \langle Z_1, Z_1 \rangle$.
The first rewriting rule is listed (among other) on Table 6.
In some cases, however, a filter cannot be completely eliminated. Below are two examples ($|Z_1| > 1$):

a) $[X_1 = X_2]\langle S - Z_1, S - Z_2, Z_3 \rangle \stackrel{\text{Tab. 6}}{\longrightarrow}$
$[X_1 = X_2]\langle (S - Z_1) \cap (S - Z_2), (S - Z_1) \cap (S - Z_2), Z_3 \rangle$

b) $[X_1 = (\neq) X_2](\langle Z_1, Z_1, Z_3 \rangle)$

In such cases we only have to split (expand) the tuple-arguments. After expanding terms like $(S - Z_1) \cap (S - Z_2)$ (using the rules on Table 4) there might be any terms still preceded by a filter, taking the form of b). Splitting would be required to finally eliminate (evaluate) residual filters.
Also in case of filter elimination, however, splitting dynamic subclasses of cardinality greater than 1 (whether tuples are preceded by a filter or not) would be necessary in order to find out enabled symbolic instances.

Final Expansion of symbolic tuples Since a potentially enabled symbolic tuple (potentially stands for possibly preceded by a filter) corresponds to a quotient-set (with respect to the equivalence relation between symbolic color instances), the idea is thereby to perform a clever splitting, by considering one representative for each equivalence class of the quotient-set. With simple combinatorics it is possible to quantify the quotient-set.

Cor. 3 *Let $\widehat{c}_t$: $\langle ds_1, \ldots, ds_n \rangle$ be a potentially enabled symbolic color tuple on $\widehat{m}$. Let $R = \{ds_k\}$ be the repeated dynamic subclasses on $\widehat{c}_t$ s.t. $|ds_k| > 1$, and n_k be the number of occurrences of ds_k on $\widehat{c}_t$. Then $\widehat{c}_t$ represents a quotient set of cardinality $\prod_{ds_k \in R} \epsilon_k$, where*

$$\epsilon_k = \begin{cases} 2^{n_k} - n_k & \text{if } n_k \leq |d_k| \\ \sum_{i=0}^{|d_k|-1} \binom{n_k}{i} & \text{otherwise} \end{cases}$$

An intuitive interpretation of Cor.3 is that only dynamic subclasses *repeatedly* occurring on symbolic tuples must be split in order to obtain corresponding equivalence classes of symbolic firings. A last example will highlight the meaning of Cor.3 and how it can be exploited.
Assume that the tuple $\langle Z_1, Z_1, Z_1, Z_2 \rangle$, $Z_1 \geq 2$, $Z_2 \geq 2$, has been evaluated as (potentially) enabled in SM $\widehat{m}$ during a local enabling test of t. According to Cor.3 the cardinality of the corresponding quotient-set only depends on $|Z_1|$: if $|Z_1| = 2$, then it is equal to 4, if $|Z_1| \geq 3$ then it is equal to 5. Letting $|Z_1| = 2$, the (four) equivalence classes are represented by:
$\langle Z_{1,1}, Z_{1,1}, Z_{1,1}, Z_{2,1} \rangle$, $\langle Z_{1,1}, Z_{1,1}, Z_{1,2}, Z_{2,1} \rangle$
$\langle Z_{1,1}, Z_{1,2}, Z_{1,1}, Z_{2,1} \rangle$, $\langle Z_{1,2}, Z_{1,1}, Z_{1,1}, Z_{2,1} \rangle$

Without this achievement, we should first expand a potentially enabled tuple into its $\prod_{|ds_k|>1} |ds_k|^{n_k}$ split components (we are assuming that R dynamic subclasses refer to non-ordered class), then we should check equivalences between them.
After evaluating the local enabling functions for a given transition t (one for each place connected to t via an inhibitor and/or an input arc) we only need to intersect the resulting symbolic sets to finally obtain a compact representation of symbolic color instances enabled on a given SM $\widehat{m}$. By construction such a compact representation has the form of a disjoined sum of symbolic tuples (each) representing an equivalence class of (symbolic) color instances.

EVALUATING THE APPROACH

The approach presented in this paper was applied to the WN model of a realistic system, a software Distributed Memory (DM) which is the main component of a complex fault-tolerant middleware. The DM is a scalable, multi-task component based upon software replication that implements safe read/write operations on application variables, thanks to sophisticated voting and task synchronization algorithms. We refer to [Ballarin et al., 2003] for a complete description of the DM model.
The DM portion depicted in Figure 2 represents the operations done by a master-task (a task which plays the role of authority for a given application variable) during its life-cycle: the compatibility check of DM variable replicas after a distributed voting (transition $CHEK_COMP$) and the issuing of a read request received from an application task on a given variable (transition RD_MST). Basic color classes represent task identities (C_1), application variables (C_2) and the values they assume (C_3), and some control codes (C_4). Some color classes are partitioned in static subclasses, in particular $C_{1,1}$ denotes the DM task group.

Efficiency issues

Some experimental results are reported that outline the improvements we get from using the algebraic approach to enabling test (standing alone and combined to the symbolic marking notion) over a simulation engine based on a traditional enabling test, enriched with some heuristics, like that one implemented in the *GreatSPN* package. In particular we refer to the symbolic discrete-event simulation algorithm of *GreatSPN* (that performs a traditional test on symbolic color instances), what makes significant the comparison. For that purpose we realized a prototype implementing an algebraic simulation engine, that works on ordinary as well as sym-

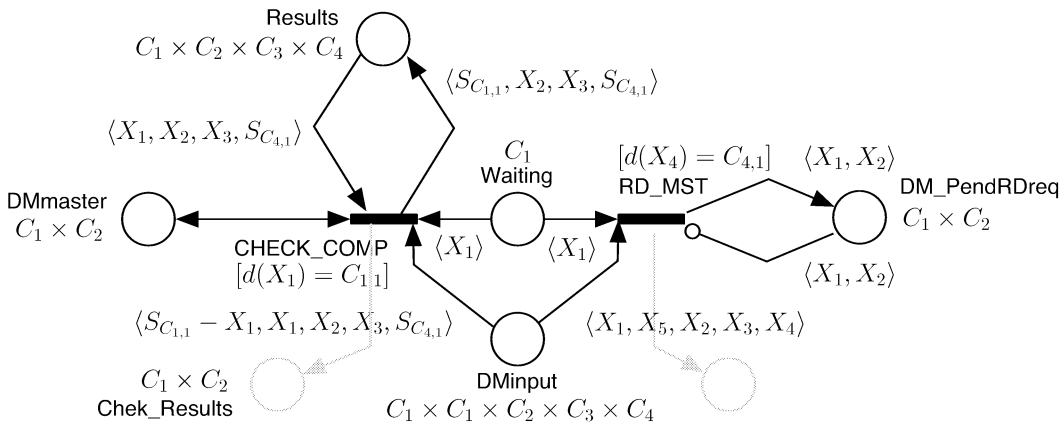


Figure 2: A portion of Distributed Memory

bolic transition color instances. The prototype takes local enabling functions as input, i.e., it assumes they have been pre-computed.

Unfortunately, for the complexity of model's color annotations, we were able to simulate a reduced number of system's configurations using the *GreatSPN* algorithms: the $dmt = 3 \times var = 4$ configuration in fact represents a threshold for both the discrete-event simulation engine and (obviously) the analytical solver of *GreatSPN* (what makes highly unreliable trying to infer tendency curves on realistic working scenarios).

Moreover, the model arc functions can be treated neither using the heuristics defined in [Gaeta, 1996] nor the approach defined in [Ilie and Rojas, 1993]. The approach that combines symbolic enabling test and SM notion, being general, moving complexity from run-time to algebraic arc function manipulation (that is performed only once), and exploiting symmetries, might be conveniently used to simulate more realistic DM configurations.

Table 2 shows the average simulation time speed-up achieved by the algebraic simulation engine over a traditional one, for increasingly complex configurations. Data on 2nd column refer to the symbolic enabling test algorithm exclusively, while data on 3rd column refer to the combined approach presented in this paper. We can thus better appreciate the contribution due to the integration of symbolic enabling test and SM technique.

That is even more evident looking at Table 2, that report data about the (maximal) percentage of memory saving with respect to a traditional approach, during execution of discrete-event simulation runs. All the experiments were carried out on a Pentium 4 630 with 2 GByte of RAM.

We can intuitively explain the good performances of the combined symbolic approach (3rd column) with two main reasons: the complexity of color domains (i.e. the "size" of symbolic tuples) and arc functions, that makes expensive testing the enabling of transitions through the traditional enabling expression, and the shape of arc functions (where projection symbols are rarely repeated), that makes splitting of source

Table 2: Simulation time Speed-Up

(dmt,var)	S.E.T.	S.E.T. + SM
(2,2)	1.2	2.3
(2,3)	1.9	4.5
(3,2)	3.9	10.3
(2,4)	4.7	9.5
(2,5)	5.1	15.
(3,3)	5.3	17.4
(2,6)	5.9	20.
(3,4)	6.2	23.8

Table 3: Percentage of Memory Saving

(dmt,var)	S.E.T.	S.E.T. + SM
(2,2)	4	5
(2,3)	4	7
(3,2)	6	13
(2,4)	7	20
(2,5)	10	41
(3,3)	15	63
(2,6)	19	88
(3,4)	21	92

SMs be only occasionally performed.

CONCLUSION

A recently developed algebraic approach to WN symbolic enabling test has been extended to work on WN symbolic markings, giving rise to a fully symbolic discrete-event simulation kernel that exploits behavioral symmetries encoded in WN models. Some experimental evidences of achieved efficiency improvements on a complex WN model of a software distributed memory (used in the past as application example of WN's algebraic enabling test) are also presented. The prototype used for the experiment is being fully implemented in the perspective of developing in the middle-term an algebraic simulation engine for (S)WN models.

REFERENCES

P. Ballarin, L. Capra, M. De Pierro, and G. Franceschinis. Memory fault-tolerance mechanisms: Design and configuration support through SWN models. In *Proceeding of the 3rd International Conference on Application of Concurrency to System Design (ACSD'03)*, IEEE, Guimaraes, Por, June 2003.

L. Capra and M. De Pierro. Efficient enabling Test in Simulation of SWN. In *Proceeding of the 20th annual European Simulation and Modelling Conference (ESM'2006)*, Toulouse, Fra, October 2006. EUROSIS-ETI.

L. Capra, M. De Pierro, and G. Franceschinis. A High Level Language for Structural Relations in Well-Formed Nets. In Gianfranco Ciardo and Philippe Darondeau, editors, *Proceeding of the 26th International Conference on Application and Theory of Petri Nets*, LNCS 3536, pages 168–187, Miami, USA, June 2005.

G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets for symmetric modelling applications. *IEEE TC*, 42(11)(11):1343–1360, 1993.

G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. A symbolic reachability graph for coloured petri nets. *Theoretical Computer Science B (Logic, semantics and theory of programming)*, 176(1&2):39–65, 1997.

R. Gaeta. Efficient discrete-event simulation of colored petri nets. *IEEE TSE*, 22(9)(9):629–639, September 1996.

J.M. Ilie and O. Rojas. On well-formed nets and optimizations in enabling test. In *Application and Theory of Petri Nets 1993, LNCS*, volume 691, pages 300–318. Springer-Verlag, 1993.

K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag. ISBN: 3-540-60943-1., 1997.

AUTHOR BIOGRAPHY

Lorenzo Capra was born in Monza (Italy), and went to the University of Milan, where he obtained his Laurea degree in Computer Science in 1992. After collaborating for a couple of years with the Automation Research Center at the National Electric Power Provider (ENEL), he moved to the University of Turin, where he received a Ph.D in Computer Science. He is currently assistant professor at the Dept. of Informatics and Communication (Di.C.O) at the University of Milan. His research interests include High-Level Petri Nets analysis/simulation and formal methods in software engineering.

APPENDIX: REWRITING RULES

Table 4: Intersection between symbolic tuples - expansion

Symbolic Operation	Rewriting Rule
$\langle x_1,\ldots,x_n\rangle \cap \langle y_1,\ldots,y_n\rangle$	$\langle x_1 \cap y_1,\ldots,x_n \cap y_n\rangle$
$S \cap x$	x
$Z_j^i \cap (S - Z_k^i)$	if $j \neq k$ Z_j^i, else ϕ
$Z_j^i \cap Z_k^i$ $(j \neq k)$	ϕ
$S_{i,k} \cap Z_j^i$	if $d(Z_j^i) = k$ Z_j^i, else ϕ
$S_{i,k} \cap (S - Z_j^i)$	if $d(Z_j^i) \neq k$ $S_{i,k}$, else $S_{i,k} - Z_j^i$
$(S_{i,k} - Z_j^i) \cap Z_w^i$	if $j = w \vee d(Z_w^i) \neq k$ ϕ, else Z_w^i
$\bigcap_{Z_j^i \in \widehat{C}_i}(S - Z_j^i)$	ϕ
$\bigcap_{Z_j^i,\, d(Z_j^i)=k}(S_{i,k} - Z_j^i)$	ϕ
Symbol	**Expansion Rule**
S (of class C_i)	$\sum_{Z_j^i \in \widehat{C}_i} Z_j^i$
$S_{i,k}$	$\sum_{Z_j^i,\, d(Z_j^i)=k} Z_j^i$
$\bigcap_{Z_j^i \in \widehat{C}'_i}(S - Z_j^i)$	$\sum_{Z_j^i \in \widehat{C}_i / \widehat{C}'_i} Z_j^i$
$\bigcap_{Z_j^i \in \widehat{C}'_{i,k}}(S_{i,k} - Z_j^i)$	$\sum_{Z_j^i, d(Z_j^i)=k \wedge Z_j^i \notin \widehat{C}'_{i,k}} Z_j^i$

Table 5: Guard evaluation without splitting

$[in] : [cl \wedge \ldots]$, $\widehat{c} : \langle ds_1,\ldots,ds_m\rangle$	Rewriting of $[in](\widehat{c})$
$cl : X_i = X_j, ds_i \neq ds_j$	ϕ
$cl : X_i \neq X_j, ds_i \neq ds_j$	$[\ldots](\widehat{c})$
$cl : d(X_i) = (\neq) C_{j,k}$	if $d(ds_i) = (\neq) k$ $[\ldots](\widehat{c})$, else ϕ

Table 6: Filter evaluation

$[out] : [cl \wedge \ldots]$, $\widehat{b} : \langle t_1,\ldots,t_n\rangle$	Rewriting of $[out](\widehat{b})$
$cl : X_i = X_j$, $t_i \neq t_j, t_i \neq Z_k^s$	$t_i, t_j \longrightarrow t_i \cap t_j$
$cl : X_i = X_j$, $t_i : Z_k^s, t_j : (S - Z_k^s) \cap x$	ϕ
$cl : X_i \neq X_j$, $t_i : Z_k^s, t_j : (S - Z_k^s) \cap x$	$cl \longrightarrow true$
$cl : X_i \neq X_j$, $t_i : Z_k^s, t_j \neq (S - Z_k^s) \cap x$	$cl \longrightarrow true$, $t_j \longrightarrow t_j \cap (S - Z_k^s)$
$cl : d(X_i) = (\neq) C_{j,k}$, $t_i : S_{j,k} \cap x \vee$ $t_i : (S_{j,k} - Z_s^j) \cap x$	$[\ldots](\widehat{b})$ (ϕ)
$cl : d(X_i) = (\neq) C_{j,k}$, $t_i \neq S_{j,k} \cap x \wedge$ $t_i \neq (S_{j,k} - Z_s^j) \cap x$	$cl \longrightarrow true$, $t_i \longrightarrow t_i \cap S_{j,k}$ $(\longrightarrow t_i \cap S - S_{j,k})$
$cl : X_i =!^k X_j$, $t_i : Z_m^l, t_j : (S - Z_r^l) \cap x$, $Z_r^l =!^k Z_m^l$	ϕ
$cl : X_i \neq!^k X_j$, $t_i : Z_m^l, t_j : (S - Z_r^l) \cap x$, $Z_r^l =!^k Z_m^l$	$cl \longrightarrow true$

EVALUATION OF VEHICLE ROUTING SOLUTIONS BY TIME PETRI NETS UNDER CONDITION OF TRAVEL TIME UNCERTAINTY

Gerrit K. Janssens
An Caris
Katrien Ramaekers
Transportation Research Institute (IMOB)
Hasselt University, Diepenbeek, Belgium
Email: {gerrit.janssens,an.caris,katrien.ramaekers}@uhasselt.be

and

Dencho N. Batanov
Department of Engineering and Science
Frederick Institute of Technology
Nicosia, Cyprus
Email: {com.bd@fit.ac.cy}

KEYWORDS

Vehicle routing, Time windows, Time Petri Nets

ABSTRACT

Congestion on roads leads to uncertainty in travel times, which is important in delivery of goods, especially in a business environment where high levels of customer service are expected. Delivery periods to customers might be constrained by time windows, which makes the scheduling and routing of vehicles from the supplier's side more difficult. Operations Research methods turn into heuristics for this type of application. But when, on top, uncertainty on travel times are the case, any hope of a simple and well-performing heuristic is lost. This study applies a methodology in which a heuristic is used to find a solution for scheduling and routing under deterministic travel times and, by means of simulation, though the use of Time Petri nets evaluates the sensitivity of the solution to uncertainties in travel times from one customer to the next in a route.

INTRODUCTION

In a world with congestion on the highways and on other roads, uncertainty in delivery times is apparent. The fact that higher service levels and shorter delivery times are expected from a customer's side to reduce the levels of inventories makes the uncertainty in travel times due to congestion more important. The planning of vehicles and their routes, known as the Vehicle Routing Problem (VRP) therefore faces an additional problem when uncertainty is involved. Various algorithms and heuristics exist for solving the vehicle routing problem in the deterministic case. The methods mostly assume deterministic travel times between customers, which might be an unrealistic assumption.

Time windows appear as constraints in scheduling problems in which there are some earliest or latest times defined on which a customer is visited or a job starts. They appear naturally in business organisations that work on fixed time schedules, including postal deliveries, waste collection, dial-a-ride service and school bus routing. In the vehicle routing and scheduling problem with time window constraints (VRSPTW) stem from the fact that some customers impose delivery deadlines and earliest-delivery-time constraints. In this application the total costs do not only include total travel distance and time costs, but also the cost of waiting time incurred when a vehicle arrives too early at a customer of while servicing the customer (loading or unloading). When the problem is enriched with time windows (either from the depot or imposed by the customers) the methods of finding optimal solutions become even harder and analysts turn to heuristics for finding a feasible solution.

This paper considers the *Vehicle Routing Problem with Time Windows* (VRPTW). Time constraints may be related to service requests. A time window represents the time interval in which the service at a customer should take place. It has a lower bound and an upper bound. Savelsbergh and Sol (1995) mention the existence of implicit time windows. These time windows refer to the desired delivery time of customers. Customer inconvenience can be controlled by taking the implicit time windows into account. Taillard et al. (1997) further distinguish between soft and hard time windows. When the time window is soft, the vehicle can arrive before the lower bound or after the upper bound. If the vehicle arrives early, it has to wait to start its service. If a vehicle is late, a penalty for tardiness is incurred. When the time window is hard, late services are

not allowed. Hard time windows are assumed. Time constraints may also be related to vehicles. In reality vehicles are not available all the time. A vehicle departs from and arrives at a single depot. In this case a time window refers to the time interval in which the vehicle is available. Both types of time constraints are referred to as customer windows and depot windows. In the VRPTW, the total cost does not only include the total travel time, but also the waiting time incurred when a vehicle arrives early at a customer, and the service time at the customer's site (loading or unloading).

The VRPTW has been subject of intensive research. As solving the NP-hard VRPTW to optimality remains hard, research has focussed on heuristic approaches. Bräysy et al. (2004) divide the heuristics in three classes. First, construction heuristics construct routes sequentially or in parallel until the vehicle's capacity is reached, without violating time window constraints. A second class consists of improvement heuristics. These heuristics try to improve the incumbent solution. Construction and improvement heuristics are discussed in detail in a first article by Bräysy and Gendreau (2005a). Finally metaheuristics are used to guide construction and improvement heuristics to escape local optima. More details can be found in a recent survey on metaheuristics in a follow-up article by Bräysy and Gendreau (2005b).

Once a heuristic solution for the vehicle routing problem is found, the solution might be evaluated against various scenarios including uncertainty in travel times. The paper first models a single route as a project network and links the notions of slack and criticality with notions in the vehicle routing solution. Further evaluation of the solution is investigated if the uncertainty in travel time can be expressed as an interval between finite lower and upper bounds. This means the uncertainty is not expressed in terms of probability distribution. The use of intervals allows for deterministic simulation of the 'what if'-type without the use of random numbers. The algorithm to produce the required output for the answers to the 'what if'-questions is programmed in a classical programming language. No specific PN-tool has been used.

The remaining part of the paper is organised as follows. First it is shown how a route, the customer time windows and the depot window can be represented as a project network. A project network can be represented by a marked graph, a special type of Petri net. When time is added to the transitions, performance of the net might be evaluated. Finally the travel time uncertainty is modelled by introducing time intervals instead of deterministic values for the transition time. Simulation of the effect of the uncertainties is done by evaluating all time intervals, in which events can take place, by means of algorithms for the performance evaluation of Time Petri nets.

NETWORK REPRESENTATION OF A VEHICLE ROUTING PROBLEM SOLUTION

This section shows that the solution of a VRPTW may be represented as a set of project networks. Each project network represents a single route. First, the modeling of routes as project networks is described and, afterwards an illustrative example is given.

A *route* is defined as a sequence of customers to be visited by a single vehicle. Each route may be modelled as a project network. The activities of a vehicle include travel and service. A route is a sequence of (travel, service)-pairs of activities. Waiting times are introduced due to the synchronisation with the customers' time windows. The opening and closing times are expressed as pairs of virtual activities in the network. A special path exists from the start node to the finish node expressing the time window of the depot, defining the earliest leaving time and latest arrival time of a vehicle. In any feasible solution this path should be the critical path (or one of the critical paths) in the network. In case the time window of the depot is not the critical path or one of the critical paths, it means that the total time of the route exceeds the time window of the depot, which does not lead to a feasible solution.

A *project network* is a directed, connected, acyclic graph $G(A,V)$. In this graph V is a set of nodes and $A \subset V x V$ is a set of arcs. Two alternative representations of project networks are available: activity-on-arc project networks and activity-on-node project networks. In this paper an activity-on-arc representation is used, which means that each activity is represented by an arc. A node represents an event.The set of arcs A consists of several subsets related to various concepts of the route and its customer. The set is partitioned as $A = (A^a, A^d, A^{ce}, A^{cl})$.

The procedure to construct the network consists of the following steps:

- Construct the backbone of the network (A^a)
- Add the depot time window to the network (A^d)
- Add the earliest customer arrival times to the network (A^{ce})
- Add the latest customer arrival times to the network (A^{cl}).

The *backbone* of the network is made up of arcs belonging to A^a. The backbone consists of alternating travel and service activities. The weights of the arcs correspond to resp. travel times or service times.

The *depot time window* is represented by a single arc A^d. The weight of the arc is equal to the opening time of the depot, under the assumption that the opening time of the depot is put equal to zero.

The *earliest service times* are represented by a set of arcs A^{ce}. The start of a service at a customer's site, earlier than the lower bound of its time window, is prohibited (in a

feasible network) by adding an arc from the depot node v_1 to the node starting a service activity at a customer's site. The weight of the arcs corresponds to the lower bound of the customer window, under the assumption that the opening time of the depot is put equal to zero.

The *latest service times* are represented by a set of arcs A^{cl}. The start of a service at a customer's site, later than the upper bound of its time window, is prohibited (in a feasible network) by adding an arc from the node starting a service activity at a customer's site to the depot node v_{2k}. The weight of the arcs corresponds to the difference between the closing time of the depot and the upper bound of the customer window, under the assumption that the opening time of the depot is put equal to zero.

ILLUSTRATIVE EXAMPLE OF THE METHOD

Figure 1 shows a small network is described with one depot and three customers {K1,K2,K3}. A route is shown in which the vehicle moves from the depot to customer K1 (10 time units travel time), to customer K2 (12 time units travel time), to customer 3 (8 time units travel time) and back to the depot (26 time units travel time).

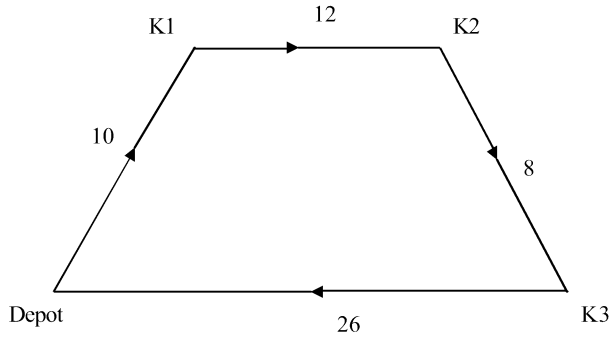


Figure 1: A single route visiting three customers

Location	*ETA*	*LTA*
K1	0	50
K2	10	50
K3	20	100
Depot	0	100

Table 1: Time windows for depot and customers

The service time at all customers is equal to 5 time units. The customer and the depot have time windows as shown in Table 1.

The route in Figure 1 can be represented as a sequence of activities. This is shown in the lower part of Figure 2. In the upper part of the same figure, the depot window is added to the network. In Figure 3 the customer windows are introduced: the earliest times of arrival (denoted as ETA) and the latest times of arrival (denoted as LTA) as given in Table 1. The latest arrival time for the customer K1 is equal to 50, which means the vehicle has to arrive before the time unit 50. The arcs, added in Figure 4, indicate the time left between the latest arrival of the vehicle and the upper bound of the depot window. For the customer K1 this means the arc has weight 100-50 = 50 time units.

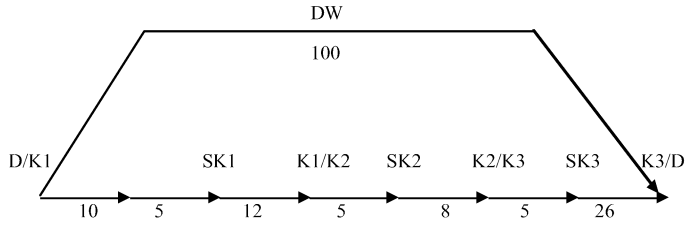


Figure 2: Travel and service activities and the Depot window introduction

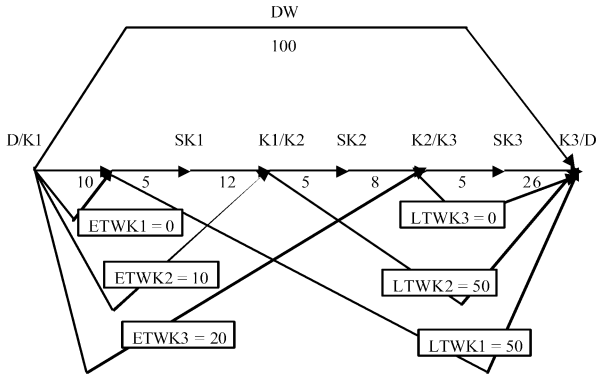


Figure 3: Customer window introduction into the project network

The first three columns in Table 2 show the information, from Figure 4, in a form as required by a project planning algorithm. The last three columns show part of the output produced by such an algorithm. Table 2 shows that activity DW, indicating the depot window, forms the critical path and is the only critical path. This means that this solution is feasible in terms of the depot window.

The project network offers a framework building scenarios for sensitivity analysis, for example by introducing other values for travel times to investigate their effect on feasibility. As an example, assume that the travel time from the depot to customer K1 takes 33 instead of 10 time units. A second critical path is found. The activities D/K1, SK1, K1/K2 and LTWK2 become the activities on the second critical path. The solution is still feasible, but several activities loose slack.

When, for instance, the travel time, due to some uncertainty, from the depot to customer K1 takes more than 33 time units, the vehicle arrives late at customer K2. There is no more slack to counter uncertainty in the travel times from the depot to customer K1 and from customer K1 to customer K2. The latest arrival time at customer K2 has become critical.

Activity	Im.Pred.	E T	ES	LS	Slack
DW	-	100	0	0	0
D/K1	-	10	0	23	23
ETWK1	-	0	0	33	33
ETWK2	-	10	0	40	40
ETWK3	-	20	0	49	40
SK1	D/K1, EWTK1	5	10	33	23
LTWK1	D/K1, EWTK1	50	10	50	40
K1/K2	SK1	12	15	38	23
SK2	ETWK2, K1/K2	5	27	56	29
LTWK2	ETWK2, K1/K2	50	27	50	23
K2/K3	SK2	8	32	61	29
SK3	K2/K3, EWTK3	5	40	69	29
LTWK3	K2/K3, EWTK3	0	40	100	60
K3/D	SK3	26	45	100	29

Table 2: Route information in a project network analysis form

INTRODUCTION OF TRAVEL TIME UNCERTAINTY

It has been shown that a single route of a solution to a Vehicle Routing Problem with Time Windows can be represented as a project network. In such a way the feasibility of the route can be tested when other values of travel times are realised. The notion of criticality is the key to this test. Sensitivity towards travel times can be performed only one customer trip at a time.

In the case upper and lower bounds on the travel times are known as an interval, an algorithm needs to be developed in which the interval is given for each travel time and the feasibility of the route can be checked. This paragraph formulates some hints how a solution to this problem may be found.

The dynamic behaviour of the route and the assessment of its feasibility in the case of uncertain travel times might be studied using the Petri net formalism. In terms of structure, it has been shown that a project network might be represented as a marked graph, which is a special case of a Petri net. Ordinary Petri nets do not include any notion of time and are aimed to model only the logical behavior of systems by describing the causal relations existing between events. A timing specification is required if performance, scheduling or real-time control needs to be considered. As this study assumes the presence of time windows and travel times, the Petri net model has to be extended with time aspects.

Two basic models for handling time have been defined: time Petri nets and timed Petri nets. A time interpretation can be associated with either places or transitions. The Time Petri net is more general (Merlin and Farber, 1976). In this case, Time Petri Nets are the most suitable class of Petri nets as they allow transitions to fire only within time intervals. A time interval includes an earliest and latest firing time, once the transition is enabled.

In Petri nets without timing information, transitions, which are enabled, can fire any moment but will not necessarily do. It is not specified when, if ever, transitions will fire, and why. The occurrence of transitions is not planned. The causal structure of transitions - pre and post conditions - is described, but the control over the occurrence of transitions is not specified by the design. The Time Petri Net case is slightly different. Each transition has a static timing interval (α_i, β_i) and a transition t_i may not fire for a period of time of at least α_i after it has been enabled, and it has to fire if it has been enabled for a period of time of β_i.

Continuity, a very specific nature of time implies that an exhaustive enumeration of states (with time-information) is not possible, as it will almost always be an infinite enumeration. Instead of handling individual states, states can be grouped in state classes. Formally, the *i*-th state class is defined by its marking M_i and its firing domain D_i, where D_i contains all enabled transitions $t_{i,j}$ with their relative firing interval $(\alpha_j < t_{i,j} < \beta_j)$. Classes are pairs (M,D) in which M is a marking and D is a firing domain. Firing rules for states classes and ways to compare classes for equality have been defined (Berthomieu and Menasche, 1983). Using these state classes, a finite representation of an infinite number of reachable states can be generated, which will be mentioned further by the class reachability graph, similar to the reachability graph of states in (non-time) Petri nets.

TIME PETRI NETS AND SIMULATION OF SCHEDULES FOR THE VRPTW SOLUTION

The Petri net, for representing the VRPTW solution as a project network, has as many starting places as there are customers (or depot) with time window constraints plus one additional starting place for representing the backbone (sequence of travel and service activities). In the schedule the transitions represent events with the meanings: distance traveled, service started, and service ended. The places between the transitions can have maximally one token. If the places contain a token, they have the meanings: transporting, waiting, and servicing.

In the paths, which model the time window progress, two transitions appear representing the beginning and starting events of the window. Between both transitions, a token in the place shows that the window is open. A second place behind the second transition indicates that the window is closed. This indicates the realisation of an infeasible schedule. Synchronisation between the partial schedule and the time window path appears as follows. The

place with meaning 'waiting' and the place with meaning 'window open' appear as input places to the transition with meaning 'service started'. If both places contain a token, the schedule is still feasible and the dynamic process can continue. The final place in the schedule represents the vehicle's return at the depot.

Place	*Meaning*	*Place*	*Meaning*
p1	Wait to start TW-K1	p14	Travelling D/K1
p2	TW-K1 is open	p15	Arrived at K1
p3	TW-K1 is closed	p16	In service at K1
p4	Wait to start TW-K2	p17	Travelling K1/K2
p5	TW-K2 is open	p18	Arrived at K2
p6	TW-K2 is closed	p19	In service at K2
p7	Wait to start TW-K3	p20	Travelling K2/K3
p8	TW-K3 is open	p21	Arrived at K3
p9	TW-K3 is closed	p22	In service at K3
p10	Wait to start TW-Depot	p23	Travelling to Depot
p11	TW-Depot is open	p24	Arrived at Depot
p12	TW-Depot is closed	p25	Route completed in time
p13	Wait to start D/K1		

Table 3: Meaning of places of the illustrative example

Trans-ition	*Meaning*	*Trans-ition*	*Meaning*
t1	Window K1 opens	t11	Start of service at K1
t2	Window K1 closes	t12	Service time at K1
t3	Window K2 opens	t13	Travel time K1/K2
t4	Window K2 closes	t14	Start of service at K2
t5	Window K3 opens	t15	Service time at K2
t6	Window K3 closes	t16	Travel time K2/K3
t7	Window Depot opens	t17	Start of service at K3
t8	Window Depot closes	t18	Service time at K3
t9	Start of vehicle	t19	Travel time K3/Depot
t10	Travel time Depot/K1	t20	Finalise route

Table 4: Meaning of transitions of the illustrative example

By executing the Petri net, two types of information are obtained. If the final place in the partial schedule is not reached, then the partial schedule belongs to a non-feasible route. If the final place is reached, then the execution shows all time intervals in which a service may be started at a customer location. Both cases have their practical applications in the physical distribution decision-making.

The Petri net contains 25 places of which the meaning is explained in Table 3. It contains also 20 transitions of which the meaning is explained in Table 4. The classes (C1..C22), which are generated by the algorithm are shown in Table 5. Each class corresponds to a marking (second column) and to a set of classes to which a transition can take place (third column) with their respective time intervals (column 4).

The class C16 in Table 5 is of importance as, if this class can be reached, it means the schedule is feasible according to all types of window constraints.

From this point, first a type of 'what if'-simulations might be produced. Assume the vehicle does not leave at time 0 (transition t9) but only at time 10. The static time interval (α_9, β_9) is no longer (0,0) but it is (10,10). Class 22 from table 5 no longer appears in the 'Next Class'-column for class C0; Class 19 no longer appears in the 'Next Class'-column for class C1, but appears as a 'Next Class'-column in C2, etc. Also the number of classes changes to 20. Of importance to the decision-maker is that, in the 'what-if' situation, a class is generated with a token arriving in p25 (route completed in time).

In a next scenario, assume the vehicle leaves at time 25. In this case no token arrives in p25. The marking without 'Next Class'-column is (p6,p9,p12,p18). The vehicle arrives at customer K1 at time 35 (in the time window of K1), but arrives only at time 52 at customer K2 (two time units out of the window), so transition t4 has fired to put the token in p6. As the token from p5 has moved by firing to p6, no token is available for firing t14 and the token remains stuck in the backbone at p18. By lack of tokens the windows at K3 and the Depot close, which means that transitions t6 and t8 fire and put tokens in p9 and p12. This schedule is not feasible with the time window constraints put by the customers.

Both scenarios are not surprising if information from Table 2 is used. The activity D/K1 (travel from Depot to customer K1) has a slack of 23 time units. So a delay of departure of 10 time units causes no problem, a delay of 25 time units does. The information on 'slack' obtained from the project network approach informs the decision-maker on changes in one activity only, but not on changes in more then one activity. With this approach, several changes can be induced and the results obtained.

Class	*Marking*	*Next Class*	*Time Interval*
C0	(p1,p4,p7,p10,p13)	C1	$0 \le \theta 1 \le 0$
		C20	$0 \le \theta 7 \le 0$
		C22	$0 \le \theta 9 \le 0$
C1	(p2,p4,p7,p10,p13)	C2	$0 \le \theta 7 \le 0$
		C19	$0 \le \theta 9 \le 0$
C2	(p2,p4,p7,p11,p13)	C3	$0 \le \theta 9 \le 0$
C3	(p2,p4,p7,p11,p14)	C4	$10 \le \theta 3 \le 10$
		C17	$10 \le \theta 10 \le 10$
C4	(p2,p5,p7,p11,p14)	C5	$0 \le \theta 10 \le 0$
C5	(p,p5,p7,p11,p15)	C6	$0 \le \theta 11 \le 0$
C6	(p5,p7,p11,p16)	C7	$5 \le \theta 12 \le 5$
C7	(p5,p7,p11,p17	C8	$5 \le \theta 5 \le 5$
C8	(p5,p8,p11,p17)	C9	$7 \le \theta 13 \le 7$
C9	(p5,p8,p11,p18)	C10	$0 \le \theta 14 \le 0$
C10	(p8,p11,p19)	C11	$5 \le \theta 15 \le 5$
C11	(p8,p11,p20)	C12	$8 \le \theta 16 \le 8$
C12	(p8,p11,p21)	C13	$0 \le \theta 17 \le 0$
C13	(p11,p22)	C14	$5 \le \theta 18 \le 5$
C14	(p11,p23)	C15	$26 \le \theta 19 \le 26$
C15	(p11,p24)	C16	$0 \le \theta 20 \le 0$
C16	(p25)	-	
C17	(p2,p4,p7,p11,p15)	C5	$0 \le \theta 3 \le 0$
		C18	$0 \le \theta 11 \le 0$
C18	(p4,p7,p11,p16)	C6	$0 \le \theta 3 \le 0$
C19	(p2,p4,p7,p10,p14)	C3	$0 \le \theta 7 \le 0$
C20	(p1,p4,p7,p11,p13)	C2	$0 \le \theta 1 \le 0$
		C21	$0 \le \theta 9 \le 0$
C21	(p1,p4,p7,p11,p14)	C3	$0 \le \theta 1 \le 0$
C22	(p1,p4,p7,p10,p14)	C19	$0 \le \theta 1 \le 0$
		C21	$0 \le \theta 7 \le 0$

Table 5: Classes generated for the illustrative example

The computational power of the method however is at its strongest when uncertainty on certain activities is induced by means of intervals. In practice, this is most relevant to the travel time activities (transitions t10, t13, t16 and t19). Assume the travel time between customers K1 and K2 (transition t13) is uncertain within the interval [12,36]. From Table 2, it can be learned that the activity K1/K2 has a slack of 23 time units, so if the travel time is within the interval [12,35] no problem is finishing the schedule in time should occur. The interval proposed here is one time unit longer, causing for trouble in the extreme case.

The method identifies a class with marking {p5,p8,p11,p17}, which means the windows are open for K2 (p5), for K3 (p8), for K4 (p11) and the vehicle is traveling from K2 to K3 (p17). The class identifies two enabled transitions: (1) t13 (if the vehicle arrives between 7 and 31 time units, the system is moved to the state 'Arrived at K3') and (2) t4 (if the arrival does not take before 30 time units, the window is closed). In the latter case, the marking becomes {p6,p8,p11,p17} of which p6 means 'TW-K2 is closed' (see Table 3) and leads to an unfeasible schedule. In the former case, the marking becomes {p5,p8,p11,p18}, which means that the three windows of customer K2,K3 and K4 are open and the vehicle has arrived within the time window of K3.

CONCLUSIONS

In the field of Operations Research, the Vehicle Routing Problem with deterministic travel times is considered a hard, discrete optimization problem. Researchers try to defined efficient heuristics to solve the problem. The problem becomes more hard if extensions to the problem are added and it is called a 'rich vehicle routing problem'. These variants include aspects of the Vehicle Routing Problem that are essential to routing of vehicles in real-life (Hartl et al., 2006).

This study offers a solution to split the difficulty of the problems into an optimization problems and an evaluation problem. The type of optimization problem, using deterministic travel times, is open: heuristics or meta-heuristics or whatever greedy method. This study confronts uncertainty in travel time with the solution from one of the optimization methods. In such a way the complexity might be reduced, while still being avoided from unrealistic situations which might lead to bad decisions.

REFERENCES

Berthomieu, B. and Menasche, M. (1983) An enumerative approach for analyzing Time Petri Nets. In: R.E.A. Mason (ed.), *Information Processing 83*, North Holland, 41-46.

Bräysy, O. and Gendreau, M. (2005a) Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science* **39** (1) 104-118.

Bräysy, O. and Gendreau, M. (2005b) Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science* **39** (1) 119-139.

Bräysy, O., Hasle, G. and Dullaert, W. (2004) A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research* **159** 586-605.

Hartl, R.F., Hasle G. and Janssens G.K. (2006) Editorial to the Special Issue on Rich Vehicle Routing Problems. *Central European Journal of Operations Research* **14**(2) 103-104.

Merlin, P.M. and Faber, D.J. (1976) Recoverability of communication protocols - Implications of a theoretical study. *IEEE Transactions on Communications* **COM-24** 1036-1043.

Savelsbergh, M.W.P. and Sol, M. (1995) The general pickup and delivery problem. *Transportation Science* **29** (1) 17-29.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J. (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* **31** (2) 170–186.

BIOGRAPHIES

GERRIT K. JANSSENS received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University (UHasselt) within the Faculty of Business Administration. He also holds the CPIM certificate of the American Production and Inventory Control Society (APICS). His main research interests include the development and application of operations research models in production and distribution logistics.

AN CARIS obtained the degree of Commercial Engineer, major Operations Management and Logistics at the Limburg University Centre in 2003. After one year of practical experience in inventory management at Reynaers Aluminium, she started as an assistant for econometrics and operations research at Hasselt University in 2004 and she prepares to acquire a doctoral degree in Applied Economic Sciences. She is a member of the research group Data Analysis and Modelling and the Transportation Research Institute of Hasselt University (IMOB). She takes a research interest in modelling intermodal freight transport networks with a focus on inland navigation.

KATRIEN RAMAEKERS graduated as Master of Business Economics, option Technology at the Limburg University Centre in 2002. Her Ph.D. work deals with the integration o simulation and optimisation as a support for complex logistics decision-making and for decision support with limited information in supply and demand. She is a member of the research group Data Analysis and Modelling and the Transportation Research Institute of Hasselt University (IMOB)

DENCHO N. BATANOV received the MSc. and Ph.D. degrees from the Technical University, Sofia, Bulgaria. He has been a full professor in the Program of Computer Science and Information Management at the Asian Institute of Technology, Bangkok, Thailand till 2005. Since then he is professor at the Faculty of Engineering and Science at the Frederick Institute of Technology, Nicosia, Cyprus. His research interests include the application of information technologies in education, management, industrial engineering and manufacturing, knowledge-based and expert systems, object-oriented software engineering, distributed systems and technologies, component and framework-based software development, and e-business. He published more than 100 papers in journals and conference proceedings, several textbooks and manuals.

SIMULATION AND ANALYSIS OF CONDITION/EVENT PETRI NETS USING SOFTWARE TOOL CESIM

Petr Novosad
Milan Češka
Faculty of Information Technology
Brno University of Technology
Božetěchova 2
612 66 Brno, Czech Republic
E-mail: {novosad, ceska}@fit.vutbr.cz

KEYWORDS
CESim, C/E Petri net, Simulation, Analysis.

ABSTRACT

This paper presents a new computer tool CESim for editing, simulating and analyzing C/E Petri nets. These nets are subclass of general Petri nets. The developed program CESim consists of a graphical editor for C/E Petri nets design and automatic and interactive simulator. The tool also provides facilities for analyzing C/E systems by case graphs, occurrence nets, synchronic distances and facts. CESim has integrated a genetic algorithm for optimal case graph layout.

INTRODUCTION

Petri nets are mathematical and graphical tool for modelling concurrent, parallel and/or distributed systems. Spreading of Petri nets to many scientific and technical areas leads to developing aid computer tools. These tools make use of Petri nets a lot easier and help them to further expand.

This paper is focused on the C/E Petri net tool CESim. CESim was developed because there was no such tool with graphical interface that provides specific possibilities of analyzing C/E Petri nets. The aim of CESim is to support education of Petri nets too.

C/E PETRI NETS

This paper assumes that the reader is familiar with the basic theory of Petri nets (Reisig 1985). The Condition/Event Petri nets are subclass of general Petri nets. Transitions are called events. Net places are called conditions because they model boolean conditions. This range restriction of places gives the C/E systems a finite state space with maximum of 2^n states, where n is a number of conditions in the C/E Petri net. The finite state space provides better possibilities for analyzing system features. Graphical representation is very similar to well known P/T Petri nets (Girault and Valk 2003). For definition of further mentioned terms events, steps, cases and C/E systems see (Reisig 1985). Among specific possibilities of analyzing the C/E Petri nets belong case graphs, processes, synchronic distances and facts.

The *case graph* is a directed graph that shows dependences between all cases and steps in the C/E system. In graphical representation nodes denote cases and edges denote steps. The case graph is one of the most valuable instruments for analyzing C/E systems.

The *occurrence net* is a cycle-free net with unbranched S-elements. Thus the elements of the occurrence net form a partial ordering. The *process* of the C/E system is defined as mapping from bounded occurrence net into the contact-free C/E system satisfying two requirements. Each slice is mapped injectively onto a case. The mapping of a T-element to an event respects the environment of the event. The restriction on contact-free C/E systems is not a problem since every C/E system can be transformed into equivalent contact-free system (see Reisig 1985).

The *synchronic distance* is an important property of the C/E systems. It is the degree of dependence between occurrences of its events, i.e. in which way the occurrence of a certain event is dependent on the occurrence of other events. The synchronic distance is a metric function.

It is possible to construct formulae of propositional logic by using conditions of the C/E system. Since conditions are allowed to change, such formulae will be true or false depending on which case the system is in. Formulae which are true in all cases of the C/E system describe invariant properties and are called *facts*.

TOOL CESIM

The tool CESim (C/E Petri Net Simulator) is inspired by existing tools for P/T Petri nets (Češka et al. 1997; Češka and Urbášek 1998). It is designed for Microsoft Windows operation system. The program is located in a single executable file without need of installation or additional libraries. The tool includes three independent views. The first view serves for a net design and a simulation. The second view shows the case graph and the third view shows the occurrence net. Contents of these views can be exported to XML file or to vector graphical format EMF. At present

this tool is used mainly for support of Petri nets education. For this reason it contains an output window that shows performed actions and other informations about the system. These informations also help to understand the evolution of the system during a simulation.

CESim provides same resources for creating all elements of graphical representation of the C/E Petri nets as modern graphical editors (Figure 1). Properties of these elements can be adjusted from floating menu bar. Besides the basic elements of the C/E Petri nets, the user has available additional graphical elements for highlighting the meaning of the modelled system.

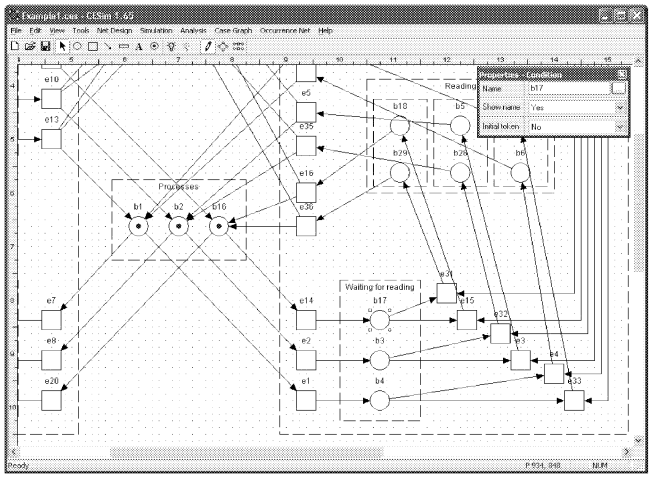


Figure 1: CESim user interface in the net design mode

The tool provides an interactive and an automatic simulation mode. The simulation works directly with graphical representation of the C/E system. The interactive simulation is entirely controlled by the user that can select and execute enabled events. The automatic simulation is driven by an algorithm with random selection of enabled events. The user can control speed of the simulation, set breakpoints on any case of the system or simulate just one step. The change of a marking during the simulation is done by animated move of tokens between corresponding conditions (Figure 2).

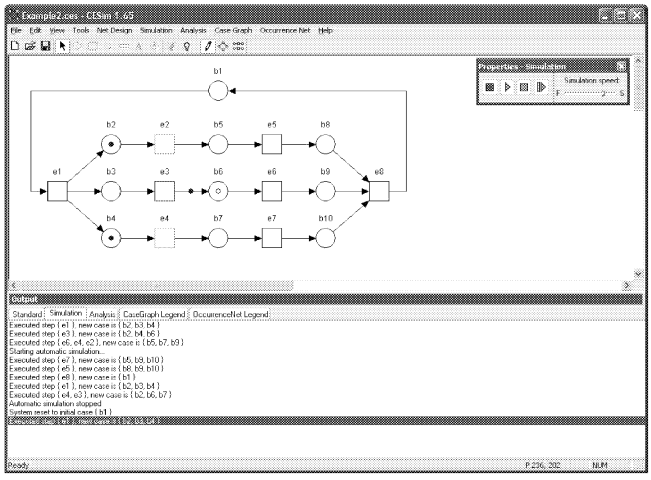


Figure 2: Animated move of a token during a simulation

The case class verification option is also available in the simulation mode (Figure 3). The user can there look up cases with certain constrains. It is possible to define which conditions must satisfy and which ones can not.

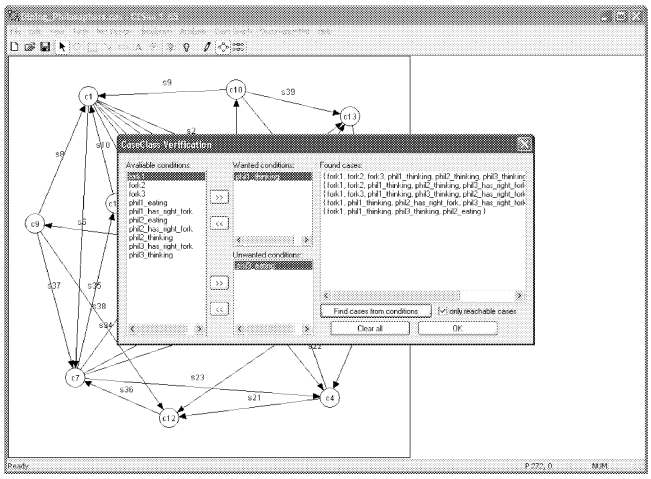


Figure 3: Verification of a C/E system

The case graph is generated automatically when the simulation mode is entered (Figure 4). The user can then additionally change positions of the case graph nodes. The graphical representation of the case graph slightly differs from the representation introduced in (Reisig 1985). The tool CESim renames labels of nodes and edges of the case graph with much shorter abbreviations. This visualization is more suitable for computer processing. Original labels are listed in the output window legend. The case graph fully cooperates with the simulation and the progress of the simulation can be controlled from it.

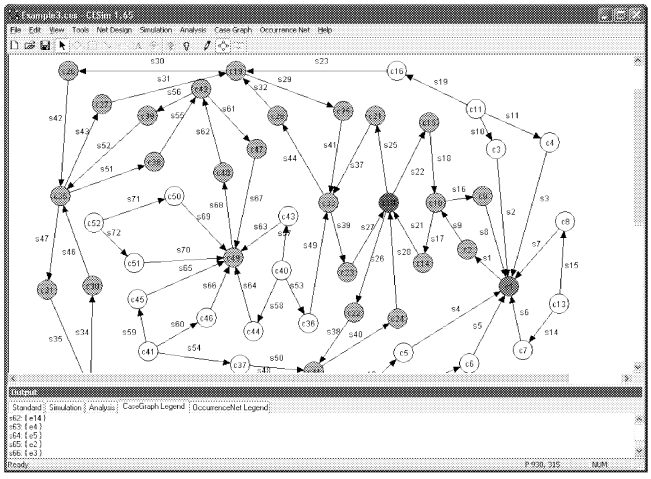


Figure 4: The case graph in the simulation mode

The case graph can be automatically drawn with minimal intersections between edges by starting an optimalization process (Figure 5). The user can set parameters of the optimalization and can control its duration while watching the displayed actual number of intersections between edges.

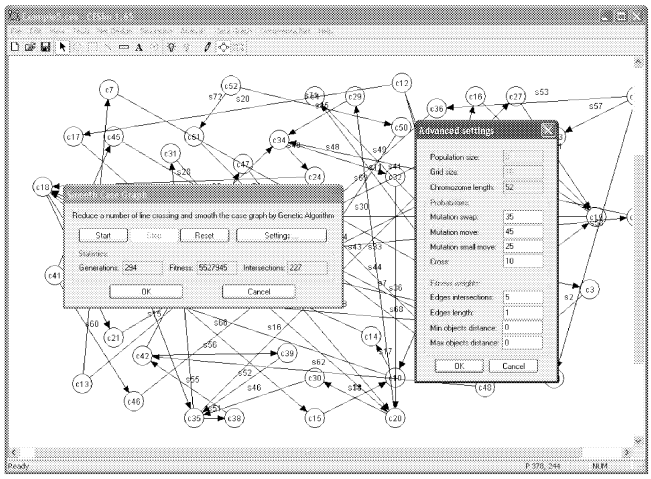


Figure 5: Smooth case graph control and settings

The occurrence net is automatically constructing during simulation (Figure 6). The user can then additionally change positions of nodes. Labels of nodes are renamed in the same way as in the case graph. Since the occurrence net is defined only for contact-free C/E systems CESim can automatically transform every C/E system to the contact-free C/E system.

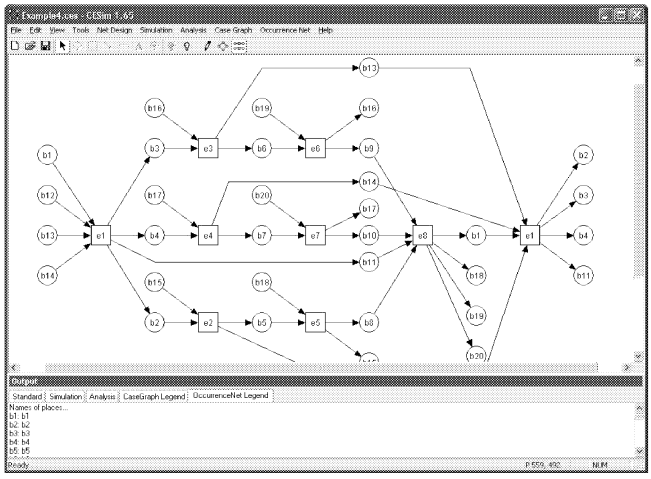


Figure 6: The occurrence net in the simulation mode

The tool CESim supports semi-automatic calculation of the graphical synchronic distance between selected events during simulation.

When simulation mode is entered, user added facts are checked for satisfying the definition of facts and corresponding formulas are written into the output window.

IMPLEMETATION

The tool CESim is implemented in developing tool Microsoft Visual C++ 6.0 using the MFC library (Prosise 2000). The size of the source code is almost 19 thousands lines. The program is object-oriented. User interface uses single document type document-view architecture. This architecture detaches document's data from its view and brings transparency to a class responsibility for a data.

The majority of C/E systems have very complex case graphs. CESim has integrated a genetic algorithm (Kvasnička 2000) for optimal case graph layout. Genes in a chromosome encode node coordinates in the auxiliary square grid. The number of genes in the chromosome equals the number of nodes in the case graph. The fitness function evaluates the number of intersections between all edges mutually and the length of edges between linked nodes in the case graph. The advantage of the genetic algorithm is in possibility of exact time control of duration of layout optimalization.

CONCLUSION

The developed tool CESim is fully comparable with similar tools in its category and moreover exceeds other tools in possibilities for analyzing the C/E Petri nets by the case graph, the occurrence net, the synchronic distance and the facts. The tool can be used for subsystems analysis aimed at improvement of their reliability and safety.

CESim assists students to get familiar with Petri nets. For this reason a set of demonstration examples were created. The students can also elaborate projects using the tool directly. Classification of students' projects is now a lot easier as well.

Future development lays mainly in implementation of support for cooperation with other Petri net tools as for example support for PNML interchange format for Petri nets.

ACKNOWLEDGEMENTS

This work has been supported by the Grant Agency of Czech Republic, grant No. GA102/05/H050 "Integrated approach to education of PhD students in the area of parallel and distributed systems", the grant No. GA102/07/0322 "Advanced Formal Approaches in the Design and Verification of Computer-Based Systems" and the Research Plan No. MSM 0021630528 "Security-Oriented Research in Information Technology".

REFERENCES

Češka, M.; Janoušek, V. and Vojnar, T. 1997. "*PNtalk – A Computerized Tool for Object Oriented Petri Nets Modelling*". In *Proceedings of the 5th International Conference on Computer Aided Systems Theory and Technology.* Las Palmas de Gran Canaria, ES, 229-231.

Češka, M. and Urbášek, M. 1998. "*Extension of the PESIM Simulation Tool*". In *Proceedings of the XX-th Workshop ASIS.* MARQ, CZ, 81-86.

Girault, C. and Valk, R. 2003. *Petri Nets for Systems Engineering.* Springer-Verlag, Berlin.

Kvasnička, V. 2000. *Evolution algorithms.* STU Bratislava, Bratislava.

Prosise, J. 2000. *Programming Windows with MFC.* Computer Press, Brno.

Reisig, W. 1895. *Petri Nets - an Introduction.* Springer-Verlag, Berlin.

PERFORMANCE ANALYSIS OF A PHOTONIC MANHATTAN STREET NETWORK USING COLORED PETRI NETS

Ajay Anand, BVR Reddy, Navin Rajpal
Guru Gobind Singh Indraprastha University,
New Delhi, India.
E-mail:anandajayanand@rediffmail.com

KEYWORDS
Performance Analysis, Multi-hop Networks, Deflection Strategies.

ABSTRACT

The requirement of high speed and high capacity network elements in future networks fully justify the interest in photonic switches and multi-hop networks. Several designs of node architectures for photonic switches are proposed and are under experimentation. One such architecture has the header of the multi-wave signal(packet) stripped off by a Wavelength Division Multiplexing filter , for decoding and processing by the routing logic. Colored Petri Nets are a graphical and mathematical tool to model and observe the behavior of a system which is complex and dynamic. This paper describes and simulates optical multi-hop Manhattan Street Network with photonic nodes, using Petri Nets.

INTRODUCTION

Introduction of fiber in the networks has brought the electronics bottleneck to the fore-front. Fibers have better throughput then the switching electronics. Optical networks, of the future, are likely to use photonic switching to avoid frequent optic-electrical conversions along the switched path. It will then be possible, to route an optical packet by the functions of the optical layer. The conversion from the optical domain to the electrical domain occurs only at the end points and switching through the network is photonic, though the switching control has to be electronic. An optical node, configured in the Manhattan Street Network, having two inputs and two outputs using radio frequency as Frequency Header labeling has been described. These nodes can be characterized as optical packet switches (OPS) and their performance has been experimentally demonstrated.

Parallel path networks, with switching nodes capable of taking decisions based on simple deflection rules, have been studied to improve the throughput. Manhattan Street Network(MSN) and Shuffle net are such networks where packets may travel through intermediate nodes to reach their destinations and are also termed as multi-hop networks.

The MSN is a two degree interconnected, mesh configured network with regular structure. It operates as a packet transmission system and the nodes are connected by two unidirectional links with adjacent rows and columns traveling in opposite directions. Logically, these wrap-around links have the effect of placing the grid on the surface of a torus so that there are no corners. Classically, even number of rows and columns are chosen and are kept equal. It is well to point out that applications of the MSN include both as MANs and as switch fabrics.

Petri Nets are graphical and mathematical modeling tool which can be applied to many systems for studying their dynamic performance. They are used for describing and studying information processing systems that are characterized as being concurrent , asynchronous, distributed parallel and/or stochastic, Colored Petri Nets(CPN) can be used to formulate methods that are well suited for modeling and analyzing large and complex system. Performance studies in the areas of Web Servers and ATM network algorithms, have used CPNs in the past The strength of the CPN lies in its capability to support both functional and performance analysis. A general purpose simulation tool like Arena provides excellent support for analyzing various types of networks. However, it does not support the analysis of the functionality of the network. In section II of this paper, we describe the network; in section III we present the CPN model of MSN network and the simulation results .

THE PROPOSED NETWORK

The MSN's with 16/36/64 nodes with Age Deflection as a routing mechanism were studied. Networks with more number of nodes can also be modeled using the suggested Petri Net model; however networks with 16/36/64 nodes are presently studied. The nodes are connected through separate optical fibers. Packets with header carrying the source and destination addresses are routed through the network. The simplicity of separating the header, which is smaller than the payload, and its subsequent processing is of immense significance. Converting the complete optical packet to electrical domain will adversely affect the processing time; instead only the header is converted to electronic domain for switching decision and is reinserted before the packet is routed onwards. Each node has two inputs and two outputs which are connected to the adjoining nodes, in the same row and column, as per the MSN configuration .In addition, nodes can leave/enter the network at each node. Packets arrive at each input and are routed to the output as per the routing algorithm. In case, in a particular time slot, there is a vacancy i.e. there is one or zero incoming packet;

one or two packets from the packets arriving at the local node and waiting in the queue is inserted. Thus, the packets at each node wait in the queue for an empty slot. The average number of Hops that a packet takes to destination, the average service time and the maximum load each node can take before saturation are some of the parameters that are studied. The packets are assumed to be of fixed length (1kbits) with 10% of the packet size as the header.

The Optical Switch Node Architecture

The optical node architecture follows the same approach as described by Fayomi and Jayasumana. The header of the multi-wave signal(packet) is stripped off by a Wavelength Division Multiplexing filter, at each node, for decoding and processing by the routing logic. The routing logic sets up the switching arrays inside the node according to the routing algorithm. The header is then optically encoded and reinserted into the rest of the packet. Each node can receive a packet on each of its inputs and can send a packet on each of the two outputs in one time slot. The node can insert one or two packets, from its local source, in case it has one or zero packet to route in a slot. The packet is delayed by one packet time in the switch by optical buffering. Optical buffers are implemented using delay lines which are fixed-length fibers. The delay of one packet time is sufficient to process the header, carry out the routing algorithm and reinsert the header for onward transmission. This delay of one time slot can be provided by optical buffer at network inputs to each node.

The Physical Link

The nodes are connected through a dedicated optical fiber link in conformation with the MSN topology.

THE CPN MODEL

Colored Petri Nets (CP-nets or CPNs) provide a framework for the design, specification, validation, and verification of

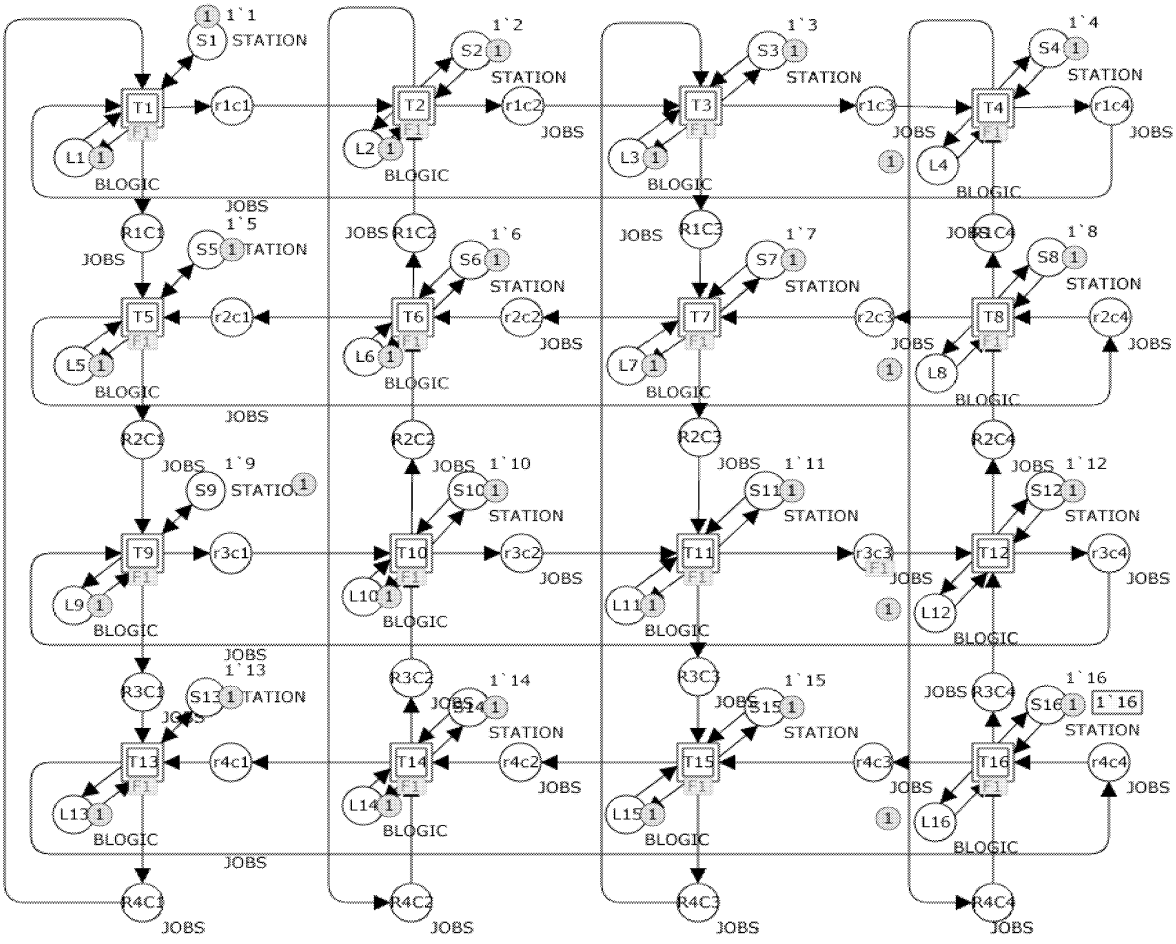


Figure 1: 4x4 MSN Petri Net Model

dynamic systems. In contrast, to most specification languages, Petri Nets are simultaneously, state and action oriented. This implies that it explicitly describes, at any step, the system state and all the possible actions the system can undertake. Petri Nets are made of Places, Transitions, Arcs and Inscriptions. The state of the system is depicted as the marking on various places which can hold data objects of a particular color (data type). A transition gets enabled, as and when the places connected to it, with output arc have markings similar to the inscription on output arcs. A firing of the transition changes the markings of the places connected to it with input arcs. The arc inscriptions and the action on firing of transitions can involve evaluation of code segments. This feature provides the Petri Net with immense power of

computations. In a given marking, the system can take an action from the set of all possible actions and as a consequence suffer a change in state, giving rise to a fresh set of possibilities. The system can choose an action from many of the equally probable actions while in a particular state. This has a potential to cause inconsistency in repeated simulation. This problem has been solved by synchronizing conflicting operations with odd/even time clock in the model being presented. Fig.1 depicts the Petri Net model of the 4x4 MSN which has user defined colors (data types) of STATION, JOBS and BLOGIC. The STATION color is unique for each node. The color JOBS has the characteristics of a packet and the BLOGIC stores the routing vector from a node to other nodes. The routing vector in case of the 4X4 MSN was manually computed and stored. However, the routing vectors for 6x6 and 8x8 MSNs were computed by the model and stored as a list, to be used to decide between the row and column on which a packet will be routed. The routing vectors are calculated as per Rule 1 suggested by Maxemchuk. Each node of the MSN is represented as a transition, in the Petri Net, with links to adjoining rows and columns The Petri Net model has an hierarchical structure wherein all the actions at the switch node are modeled on a separate page(called switching page ,in this paper) with clearly defined inputs and outputs for incoming and outgoing links . The developed switching fabric Petri Net is used as an instance at each node of the MSN. This draws upon the reusability feature of the CPN tool. The Petri Net of the switching fabric can be reused for any configuration of the MSN model. In the present model, the routing decision at the node is done based on the shortest distance to the destination algorithm. The model can be configured for any number of nodes. The nodes are numbered sequentially and the number of a particular node is passed to the connected switching page instance by marking on the place of color STATION in the model of Fig 1. This information is required by the routing algorithm to decide whether the packet needs to be passed to the network or to the local host. Each node generates traffic as per a Poisson process with a programmable mean rate of arrival. More than one packet may arrive in a particular slot. Thus, the arrival process models asynchronous arrivals. The packets generated at each node are statistically independent of each other. The model supports generation of traffic with same or different parameters at each node. The packets can form a queue and wait to be picked by the node when it finds a vacancy in the optical packet stream. If the packets on the two incoming links contend for routing on the outgoing row/column, then the packet with larger stay in the MSN is given priority. The network delay is defined as the time packet spends in traversing the network from source to destination, excluding the time spent in the input buffers. For a 64 node MSN, under uniform destination distribution assumption, the average distance between source to destination is 5.02 and a single deflection due to conflict may increase the distance by four. The network delay, in the presented model , is measured as number of hops a packet takes to reach destination. The delay on the optical links can be easily programmed in the model, however in the present study the delays on the links are neglected. Two clock periods of the CPN simulator are mapped to one network time slot. This, has been done to ensure correct sequence of operation in the switching page of the Petri Net. This circumvents the inherent limitation of the Tool which does not priorities among transitions, which are enabled in a marking of a particular state of the network and can cause potential hazard of inconsistency. The model incorporates a programmable parameter of maximum hops in the network to avoid permanent cycling of the packets. This is necessitated due to adoption of the Age Deflection strategy in the event of conflict. Monitors are attached to appropriate places and transitions to observe various parameters of the MSN.

.

RESULTS OF THE SIMULATIONS STUDY

The Petri Net model was simulated with the CPN tools available at www.daimi.au.dk .The Petri Net was simulated for various Poisson rates of mean arrival at each node. The growth of the queue of packets waiting to be inserted in the network, the average number of hops made to reach the destinations and the throughput of generated traffic at each node were observed. The packets after being generated, as per the Poisson rule, wait in the queue to be picked up by the routing network. It was assumed that the packets of 1 k bits switched through the network. As the Poisson rate of the packet generated at each node was kept the same, observation at a particular node represents occurrence at all other nodes. The simulations were run on PC with 1 GB RAM.

Average number of hops under various Network loads

The mean inter-arrival time (λ), in slot times, of the incoming packets was varied and the average number of Hops (μ) made by the packet to reach the destination for 4x4, 6x6 and 8x8 MSN are given in Table I, II and III respectively .The maximum number of hops before the packet is discarded was set to twice the total number of nodes in the network. This avoids indefinite circulation of a packet in the network.

Average Queue length of packets at the Input

The average length of the queue (α) grows rapidly after the packet inter-arrival time is reduced below 2.28, 3.58 and 4.54 in case of 16, 36 and 64 nodes MSN respectively. The inter arrival time of 4.54, for 64 node MSN, at which the network saturates and the length of the queue grows with time has also been reported as the network saturation throughput by Robertazzi and Huang. The queue length at various mean inter-arrival times, close to saturation, of incoming packets for 16, 36 and 64 nodes MSN are tabulated in Table I, II and III respectively.

Maximum load at input

Each node is connected to two input and two output links. The link load capacity of these nodes is fixed. As the traffic on the node is increased, the link load rises and reaches its maximum value when the links saturate and are unable to support any further increase in the input traffic. As the input traffic reaches this saturation value, the queue formation starts at the input node and the packet has to wait longer in the queue before being picked for routing through he MSN.

The waiting time in the queue, and the time (β) taken ,by the packet, to be picked for routing after reaching the head of the queue were studied. It was found, that for those values of λ for which the waiting time of the packet after reaching the head of the queue were less than the mean inter arrival time, the queue length did not grew with time . It was found, that the minimum mean inter arrival time of packets for which the input queue length did not grew with time for 8x8, 6x6 and 4x4 MSN was 4.54, 3.58 and 2.28 time slots respectively.

Table 1: Average Number of Hops to Destination, Average Queue Length and Average Service Time of the Head of Queue Packet for 16 Nodes MSN

λ	2..15	2.2	2.25	2.28	2.30
μ	4.19	4.19	4.125	4.109	4.13
α	*	5.47	5.25	2.628	1.807
β	3.34	2.80	2.35	2.22	2.137

* The Queue length increases very sharply with time.

Table 2: Average Number of Hops to Destination, Average Queue Length and Average Service Time of the Head of Queue Packet for 16 Nodes MSN

λ	3.00	3.55	3.58	3.60	3.65
μ	6.93	6.65	6.61	6.52	6.49
α	*	3.67	3.17	2.73	2.00
β	6.29	4.03	3.56	2.87	2.80

* The Queue length increases very sharply with time

Table 3: Average Number of Hops to Destination, Average Queue Length and Average Service Time of the Head of Queue Packet for 16 Nodes MSN

λ	4.00	4.54	4.60	5.00	5.50
μ	8.65	8.30	8.31	7.79	7.37
α	*	2.61	2.16	1.13	1.05
β	7.72	4.48	3.72	1.56	0.804

* The Queue length increases very sharply with time

A maximum value of rate of traffic, before onset of infinite growth of input queue, at each node was observed as .22(1/ 4.54), .27(1/ 3.58) , .44(1/ 2.28) in case of 8x8, 6x6 and 4x4 MSN respectively . This value though seems to be low, but considering the speed at which the network operates; even this can mean a very high traffic rate. The load on the links between the node is many times the input traffic rate,, as a packet makes more than one hop before reaching its destination.

Throughput

The throughput at each node is calculated from the difference of the outlet and inlet link loads. Note, from Little's law,

$$\text{Network Delay} = \frac{2 \times \text{Link Utilization}}{\text{Throughput}}$$

The Network Delay is measured as average number of hops taken by a packet in the network before reaching the destination; Link utilization can be obtained as a ratio of the occupied slots on the outlet link to the total number of slots in a given time; and throughput at node is the average rate at which the packets originated at the node are accepted by the optical stream and routed out of the node. Figure 2 gives the throughput at a node for various arrival rate of traffic. It is observed that the throughput initially rises linearly with increasing arrival rate, till it reaches a saturation value.

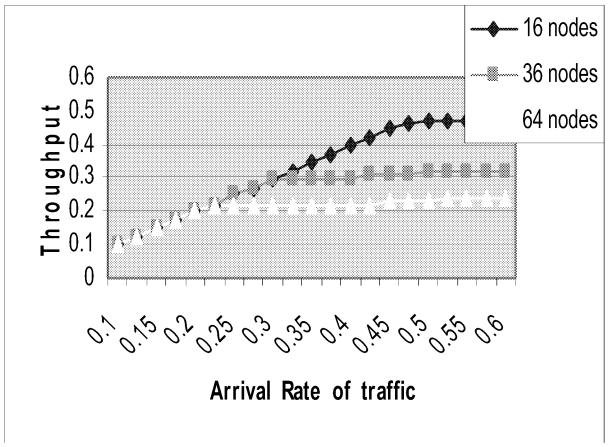


Figure 2: Throughput at a node for various arrival rate of traffic for 16/36/64 node MSN

CONCLUSIONS

We used the Colored Petri Nets (CPN) to evaluate the performance of the MSN with optical buffering and age deflection rule. The network is based on a novel optical node which provides separation of the header from the payload. The header is converted to the electronic domain while the payload is stored for unit time in optical buffers The Petri Net model of the network clearly depicts the state and action of the network under study. The model permits pausing after convenient number of steps to examine the state and possible actions of the network before resumption of the simulation process. Also the path of packets in the simulated network can be easily monitored. The CPN model exhibits good level of scalability of the network as a Petri Net of the switching fabric is used as an instance in each node of the network. The packets , in the present network, wait at the nodes ; as they can be inserted on the network only if the node of degree 2 in the MSN finds a vacancy. As the traffic intensity is increased, the throughput reaches saturation beyond which the input queue length and time taken to reach the destination increases sharply with time. Also , the time taken to enter the optical stream after reaching the head of the queue , by the packet being inserted at a node ,is greater than the inter-arrival time of packets generated at the node. This clearly depicts the growth of the queue with time causing instability and heavy cell loss Hence the network can be operated at loads below these saturation values The packets from a source can take alternate rotes to a particular destination The network model understudy is proposed to be modified to carry ATM traffic, in which the packets must take the same route from a source to destination, in the future study.

REFERENCES

(1) Barbosa, F.R.; D. Maia Jr; E. Moschin; L. Pezzola; and A.C. Sachs. 2003. "Optical Packet Switching and Routing using RF Frequency Header Labeling for Application in Metropolitan Networks", *Proceedings of SPIEITCom 2003 (Orlando ,USA,)*, 5247-320.

(2) Barbosa, F.R.; D. Maia Jr; E. Moschin; L. Pezzola; A.C. Sachs; and M.R. Salvador. 2003. "Optical Packet Switching Node for Metro-access Networks", *Proceedings of European Conference on Optical Communication,ECOC03(Romaine ,Italy)* We4.160.

(3) Clausen, H.; and P.R. Jensen. 1994. "Analysis of the Usage Parameter Control Algorithms for ATM Networks," *In Broadband Communications ii 9C-24 ,* Elsevier Science Publishers, 290-310.

(4) Fayomi, A.G.; and A.J. Jayasumana. 2001. "Effects of Optical Buffering on the Performance of Manhattan Street Networks", *Photonic Network Communications"*, 161-176.

(5) Guedes, B.C.L.; M.M Mosso; H.J.JP.P. Silva; R.A.A. Lima.; and A. Podcamani. 2005. "Performance Analysis of an Optical Switched Manhattan Street Network", *Photonic Network Communications*, 10.3, 361-370.

(6) Harari, H.; and M. Murata. 2006."Optical Fiber-Delay-Line Buffer Management in Output–Buffered Photonic Packet Switch to Support Service Differentiation", IEEE *Journal on selected areas in Communication, Vol 24 , No 8* ,1008-1161.

(7) Kelton, W.D.; R.P. Sadowski; and D.A. Sadowski. 2002. "Simulation with Arena", McGraw Hill, 2nd edition.

(8) Kristensen, M.; S. Chistensen and K. Jensen.1998. " The Practitioner's Guide to CPN's,"*International Journal on Software Tools for Technology Transfer, Springer Verlag" ,* 98-132.

(9) Lee, W.T. 1995. "Binary Addressing and Routing Schemes in the Manhattan Street Network", *IEEE/ACM Transactions on Networking ,Vol 3, No 1* , 26-30.

(10) Maxemchuck, N.F. 1985."The Manhattan Street Network" *In proceedings of the IEEE Globecom'8 5,* 205-211.

(11) Maxemchuk, N. F. 1987. "Routing in the Manhattan Street Network", *IEEE Transactions on Communications Vol Com 35, No 5,"* 503-512.

(12) Motloy, M.K.1982. "Performance Analysis using Stochastic Petri Nets ", *IEEE Trans .Computer Vol C31, No 9C:\ Documents and Settings\ user\ Desktop\ EIGHT*,973-917.

(13) Robertazzi, T.G.; and A. A. Lazar. 1991. "Deflection Strategies for the Manhattan Street Network", *ICC91, 16 52*1658.

(14) Robertazzi, T. G.; and Hsin Yuan Huang. 1992 "Performance Evaluation of the Manhattan Street Network with Input Buffers", *ICC92,* 202-209.

.

AJAY ANAND was born in Mumbai and studied at Punjab University and Indian Institute of Sciences, Bangalore. After doing military service, he is teaching Computer communications and VLSI design.

REDDY, B.V.R. is holding the position of Professor in USIT, Indraprastha University and is member of ISTE and IETE. His teaching and research interests include semiconductor devices and circuits, microwave and optical devices and computer communications networks.

NAVIN RAJPAL is Professor in USIT, Indraprastha University and has guided research in the fields of Computer Vision, Image Processing and Algorithms Design.

A HIGH-LEVEL PETRI NET BASED MODELING APPROACH FOR RISK MANAGEMENT IN SUPPLY CHAIN NETWORKS

Gonca Tuncel
Dokuz Eylül University
Department of Industrial Engineering
35100, Bornova - Izmir, Turkey
E-mail: Gonca.Tuncel@deu.edu.tr

Gülgün Alpan
Laboratoire G-SCOP, CNRS, INPG, UJF
46 avenue Félix Viallet, F38031
Grenoble, France
E-mail: Gulgun.Alpan@g-scop.inpg.fr

KEYWORDS
Supply Chain Management, Risk Management, High-level Petri Nets, Performance Evaluation.

ABSTRACT

The aim of this study is to show, how a Supply Chain (SC) management problem can be formulated in a timed Petri nets framework, and how this model can easily be extended to cover risk management applications. In this paper, the risk management procedures are integrated into design, planning, and performance evaluation process of supply chain networks through Petri net based simulation. The developed methodology provides an environment for defining uncertainties in the system and mitigation of risk factors. We show that, the PN framework provides an easy integration of risk management modules in the supply chain model. The analysis of this enhanced model shows that the system performance can be improved using risk management mitigation and the overall system costs can be reduced.

INTRODUCTION

In today's global marketplace, the supply chain networks have become more complex with increased uncertainty (Chandra and Kumar 2000). Even though the literature of Supply Chain Management (SCM) is very plentiful, most of the existing models can only describe a restricted class of supply chains with simplifications (Chopra and Meindl 2001). Traditional modeling approaches have some limitations to handle the degree of complexity inherent to real supply chain networks. Two major tools used for the modeling and analysis of SCM are the discrete event simulation and the mathematical programming techniques. SCM based on discrete event simulation generally deals with the tactical and operational level decisions such as inventory control, material handling, layout design, and vehicle routing/scheduling, while the mathematical programming techniques are mostly used for long-term strategic decision making.

PNs, as a graphical and mathematical tool, are powerful for modelling, analysis and performance analysis of discrete event dynamic systems, and recently have gained a growing interest of researchers. They are easy to develop and to extend. Researchers have a better understanding of the dynamic behaviour and control logic of the system by means of tokens and transitions. Activities, resources, and constraints of SC networks can be represented in a single coherent formulation. Concurrent and asynchronous activities, multilayer resource sharing, routing flexibility, limited buffers, and precedence constraints can be explicitly and concisely modelled by PNs (for further reading see DiCesare et al. 1993; Desrochers and Al-Jaar 1995; Moore and Gupta 1996; Murata 1989; Zhou and DiCesare 1993; Zurawski and Zhou 1994). Timed PNs give a state-space structure representation of possible system states of a production-distribution problem, and this structure is transformed to a problem which can be solved by combining PNs execution with other techniques such as heuristic search, artificial intelligence search techniques, heuristic dispatching, and expert systems. On the other hand, mathematical foundation of PN theory also allows the combination of PNs with some mathematical techniques to analysis of structural and behavioural properties of the system.

Although the first applications of PNs mostly focused on computer systems and communication protocols, they have later been extended and applied to a broader range of problems including the modeling, analysis, and control of production systems by means of their modeling capabilities and formulation advantages. However, the literature of PN applications in the area of SCM is not plentiful. Recently, various PN models have been proposed by researchers to the modeling, structural analysis or performance evaluation of SCs. In one of the first studies which applied PNs to SCM, Viswanadham and Raghavan (2000) employed generalized stochastic Petri nets (GSPNs) to describe dynamic behaviour of supply chain networks. The authors compared two different production planning and control policies with respect to the total cost comprising of the inventory carrying cost and delay cost. Van der Vorst et al. (2000) considered SCs as a business process, and used Colored Petri nets (CPNs) to model and evaluate alternative design scenarios through PN based simulation. The proposed methodology was illustrated with a case study in food industry. In Dong and Chen (2001)' study, a modular approach based on object-oriented predicate/transition nets (OPTNs) and Computer Integrated Manufacturing Open System Architecture (CIMOSA) based process behaviour rules is used to model and analyse the routing structures of a typical supply chain process. The authors used p-invariants through system objects to verify structural properties of the network such as deadlock and overflow.

Finally, the sequences of the operations in supply chain process are analyzed by PN unfolding techniques. A process management approach for SCs based on XML-nets is introduced by Mevius and Pibernik (2004). The advantages of the proposed modelling methodology is the capability of modelling supply chain objects and the exchange of intra- and inter- organizational data in a hierarchical structure. Demand fluctuations in SCs, so-called the Bullwhip Effect, was analyzed in PN based modelling framework in Makajic-Nikolic et al. (2004)'study. Supply chain modelling using CPN tools and a beer game implementation is presented based on hierarchical coloured PNs. Chen et al. (2005) developed a new PN model for modeling and performance evaluation of SCs by enhancing deterministic and stochastic Petri nets (DSPNs) with batch places and batch tokens. Structural analysis is performed based on state equations and reachability analysis techniques. Performance evaluation of an industrial SC which is composed of three suppliers, three transporters for the suppliers, a manufacturer, and a transporter for the customer is also presented in this study. Recently, Liu et al. (2007) proposed a new approach for supply chain event management (SCEM) through dependency graph analysis and event rules. The authors presented seven basic modelling patterns that arise commonly in SCs and they used CPN tools for the implementation of the proposed modelling methodology under different strategies.

The studies reviewed above demonstrate the advantages of PNs for SC modelling and analysis via state space representation and mathematical foundation of PN theory. However, they lack a consideration for the risk management issues, and the effects of disruptive factors are not discussed in a network environment. Risk management in supply chain is a rather new topic, and hence not much investigated yet. In the literature, supply chain management, performance analysis, and risk management (RM) have generally been considered separately. Furthermore, the risks are mostly addressed from only the financial or economical perspective (Winkler and Kaluza 2006). Although some recent literature handles the risk management from the logistics point of view, these very few studies often look at the single organizations' vulnerabilities, not in an integrated framework (Jüttner 2005). Supply chain management without considering risk issues in a systemic perspective and their impact on the performance measures eventually lead to suboptimal results and inconsistent processes. A supply chain is subject to risk of failures which is mainly due to the uncertainties in the system. It is highly important to respond to disruptions due to uncertainty in a real-time manner.

The main purpose of this study is to investigate various risks inherent to supply chains and to integrate the risk management issues in modeling and real-time control of supply chain networks. In this study, we aim to use a Petri net based modeling framework to this end. This model will provide us a computerized support for decision making through tracking of material, information, and financial flow in the SC network by maintaining current status information of the entire system, and properly generating the required data. Thus, the proposed approach will help the user to define and focus on key attributes in a comprehensive framework and develop customized solutions within the integrated network model. The remaining sections of this paper are organized as follows. In section 2, we introduce an illustrative model for SCM subject to uncertainties and risk factors. We propose some risk mitigation procedures to handle risks inherent to the system. Performance evaluation under these risk mitigation actions is performed in section 3. Finally, section 4 concludes the study with some comparative study and provides potential extensions and future work.

MODELING, PERFORMANCE EVALUATION, AND RISK MANAGEMENT FOR SCs USING PNs

System Description

In this study, we analyze the dynamics of an industrial SC with respect to the risks related to the logistics operations, quality and systems failures on the performance measures. The supply chain network considered for modeling and analysis is composed of two suppliers, a manufacturer, a retailer, inbound logistics which provides the transportation between suppliers and manufacturer, outbound logistics which is between manufacturer and retailer, and *n* customers. The general structure of this SC which will be used for illustration purposes is shown in Figure 1.

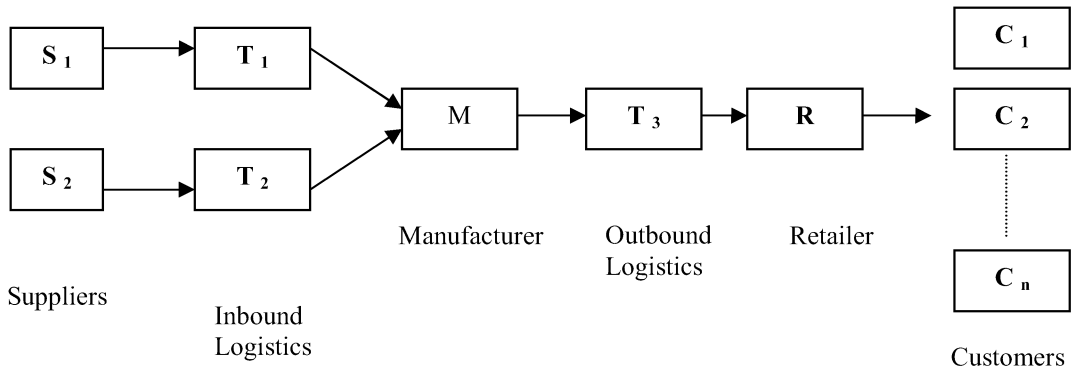


Figure 1: Structure of the Supply Chain

In our model, we will consider the suppliers (S_1 and S_2), manufacturer (M) and the retailer (R) as the major layers (actors) of the SC. The interfaces between these layers include mainly the transportation operations (T_1, T_2, T_3) and inspections of parts/products. Customers ($C_1, C_2,\ldots,C_n$) dictate the demand structure. The process starts when customer order arrives at the manufacturer with the attributes such as the arrival time to the system, due date, lot size and so on. The operational policy between the manufacturer and the retailer is under the pull paradigm. That is, if the inventory level at the retailer is enough to fulfill the order, the final products are directly sent from manufacturer to the customers through the retailer, otherwise, an assembly operation is started at the manufacturer, and a purchase order for raw materials is sent to the corresponding supplier. Raw materials A and B are provided by suppliers S1 and S2, respectively and they are delivered to the manufacturer by the corresponding transporters (T_1 and T_2). When the raw materials are received at the manufacturer site, they are controlled to check if there is any quality problem. If a raw material has a quality problem which can be solved, it is reworked, otherwise they are rejected. After the quality control process, the final products are produced by assembling part A and B. The manufacturing system is exposed to some disruptive factors that may arise from technical reliability or human errors. Once the production operation is completed, the products are inspected and if it is necessary, they are reworked. Finally, the finished products are transferred to the retailer by a transporter (T_3), where the customers can receive their products. All the transportation operations between different layers in the supply chain process are also subject to some risk factors such as damages or loss of a shipment due to accidents, or delayed arrival of a vehicle because of heavy traffic, and so on. Similarly, the manufacturing process has also some operational risks which cause the system to halt because of the breakdowns in manufacturing facilities or labor shortages.

Problem Definition

In literature, most of the applications on SC management concern small size problems where the uncertainty and risk issues are somewhat ignored. However, in real situations, the performance of a supply chain is highly dependent on the selected control policies. A SC control policy isolated from risk management issues can deteriorate the system performance. Thus, primary question is how the supply chain should be managed to capture unexpected events while maximizing the customer satisfaction and minimizing the total cost at the same time. The objective of this article is to propose a design methodology of a well-suited control mechanism and systematic decision-making process of a SC under the influence of disruptive factors. This methodology will be used to improve the overall system performance.

Taking Into Account the Risk Management in SC

A typical process of risk management contains four basic steps; risk identification, risk assessment, decision and implementation of risk management actions, and risk monitoring (Hallikas et al. 2004). These basic steps apply to risk management in supply chain processes as well. In a SC, the first step of risk management process is the risk identification which helps to develop a common understanding of the future uncertainties surrounding the supply chain, thus recognizing the potential risks in order to manage these scenarios effectively. In this study, we consider the following common supply chain disruptions: (1) delivery fluctuations, (2) interruptions at the assembly process, (3) quality failures at the suppliers and the manufacturer, and (4) fluctuations in demand. The second step includes the assessment of the probability and the consequences of a risk event defined in the first step. The company's own experiences, the other companies' performance results or forecasting analysis can be utilized to build risk diagram of the company. In this study, we assume that the occurrence probabilities of potential risks are known from historical and statistical.

After the identification and assessment of potential risks in a network environment, and understanding where to focus on, risk management decisions and actions are to be applied (Mussigmann 2006). These actions include, in general, the risk taking, the risk mitigation, the risk avoidance and the risk transfer. Here we will concentrate only on the risk mitigation scenarios which have the vocation of improving the reliability of the processes. For instance, they may represent the crew training for a reduced human error probabilities, shorter maintenance intervals or new equipment for both transportation and manufacturing system for a higher technical reliability, or higher level of safety stock (Nedeβ et al. 2006). The system performance is periodically observed and according to the levels of related performance measures, the proper strategy is selected to avoid or reduce the risk in the process and to improve the system performance. The abstraction of these actions is presented via 3 mitigation actions listed below:

- ***Mitigation action 1 :*** If the total rate of delayed and cancelled orders is higher than % 30, increase the inventory level on the average by % 15 and decrease the transportation and quality risk by %10, system failure risk by % 15, and increase the total cost by 10.000.

- ***Mitigation action 2 :*** If the proportion of delayed and cancelled orders is between %15 and % 30, decrease the transportation and quality risk by % 5, system failure risk by %10, and increase the total cost by 5.000.

- ***Mitigation action 3 :*** If the proportion of delayed and cancelled orders is between %5 and % 15, decrease the quality risk by % 5, system failure risk by %10, and increase the total cost 2.500.

The rules above can only be applied once because of the system capacities and lower limit of the occurrence probabilities of the failure risks. If the proportion of the orders which is not fulfilled before due date is still between the same limits, the next rule which is not applied yet is implemented.

Risk mitigation actions reduce the initial impact of a risk on a task, but the system will be charged with additional costs not included before. In order to evaluate the value of the integration of risk management issues in the SC model, we will use a fourth scenario which we will call the *reference scenario*. This scenario reflects the case where the system is exposed to the risks but no action is taken.

Here we consider global mitigation actions which are defined for the whole supply network. The interactions and cause-and-effect analysis of different risk factors can be dynamically examined in an integrated framework. In this context, this approach can be considered as a manner for inter-organizational risk management in supply chains. Since the dependency between the entities in a supply chain network increases considerably, it is fruitful to divide risks among partners, and the concept of risk sharing becomes an issue of supply chain risk management (Kersten et al. 2006). Given the output of the global risk mitigation actions taken in the inter-organizational level (i.e. the overall supply chain), one may perform further tests of local mitigation actions on the company level to obtain performance measures for a single company that belongs to the supply chain.

PN Modeling of the SC Network Including Risk Management

In order to formalize SCs, then to study the impact of different risk factors and mitigation actions on the system performance, we have chosen high-level PNs as a modelling and analysis tool. Petri Nets as a graphical and mathematical tool provides a uniform environment for modelling and analyzing of concurrent, asynchronous, distributed, parallel, and deterministic or stochastic systems. PN based simulation helps to conduct risk assessment and mitigation by analyzing the cause and effect relations between events and the effects of risk factors on performance measures for various scenarios. Hence, both task status related events and events produced by a task are managed synchronously to observe the effects of risk mitigation parameters. Risk mitigation is applied by executing a set of actions in order to manage the identified risk in the system according to a risk-based structure where occurrence probabilities, the impacts, and mitigation action are present. The disruptive factors are expressed in the PNs formalism by modifying the model with additional transitions characterized by a probability of occurrence. Then, scenario based heuristic rules which are developed for risk management, are also incorporated into the PN model. Mitigation actions will reduce the probability of a risk on a task, but the process will be charged with additional cost. The rule-based system developed for risk management in the SC network under consideration is constructed using the following heuristics:

- If an order is delayed, contact the customer and take the due time tolerance for the delayed order. If the current time is already out of the tolerance, cancel the order with backlogging cost.
- If the delayed product order is in the due time tolerance, send it back to the place which stores the tokens represent the customer orders waiting to be fulfilled, with higher priority.

These heuristic rules can easily be formulated using high-level PNs, which allow arbitrary complex data types for tokens. The expressions for places and transitions can be constructed using standard programming languages. Similarly, the mitigation scenarios can be modelled by increasing the system capacity or reducing the probability of corresponding risk factors, while increasing the systems operational cost in the PN based simulation model. The performance measures are collected on an event based manner (i.e. when each order is fulfilled or cancelled), and periodically (i.e. during the execution of the mitigation actions).

In this study, Artifex PN software tool (RSoft Design Group, Inc. 2002) is employed to model the system. Artifex exploits a graphical language, which is based on extended Petri Nets, can define detailed models of a system or process, and can be coupled with standard programming languages like C and C++. Figure 2 shows the Petri net model of the SC system under consideration. Tables 1 through 3 summarize the general execution process and the descriptions of the PN model.

Table 1: Interpretations of the Transitions at Supplier Part

Transition name	Description
SR_1	Starting of inventory replenishment in supplier 1
RT_1	Raw material transportation to the supplier 1
RT_R1	Raw material transportation risk
SP_1	Processing by supplier 1
ST_1	Transportation of the parts A from supplier 1 to the manufacturer
ST_R1	Transportation risk between supplier 1 and the manufacturer
SR1	Arrival of parts A to the manufacturer from supplier 1 and quality inspection
SQ1	Receiving of parts which have no quality problem
SQ_R1	Receiving of parts which have quality problem
QR_1	Reworking of parts
QF1	Rejecting of the parts which have unsolvable quality failure
SET_INV_1	Setting the inventory level of the supplier 1

Table 2: Interpretations of the Transitions at Manufacturer

Transition name	Description
M1	Starting of assembly operation by manufacturer
M2	Assembly operation
M3	Receiving of the assembled products which have no quality problem
PQ_R1	Receiving of the products which have quality problem
PQ_R	Reworking of the products
PQ_R2	Rejecting of the products which have unsolvable quality failure
M_F	System failure risk
PF_R	Resolving the system disruptions
SET_INV_3	Setting the inventory level of the manufacturer

Table 3: Interpretations of the Transitions at Retailer

Transition name	Description
R1	Transportation from manufacturer to retailer
PT_R	Transportation failure risk between manufacturer and retailer
D1	Arrival of customer orders
R2	Delivering products to the customer
O_BACK	Checking due date status of orders and send back the rush orders with high priority
P8	Notifying the customer about the order delay, and assigning due date tolerance to the corresponding tardy order
O_CANCEL	Cancellation of the orders which are already out of the due date tolerance

PERFORMANCE EVALUATION

In addition to giving the user insight into how complex systems function and how variables interact with each other, the proposed model provides the user with an informative approximation of "what-if" scenarios to assist in supporting operational and real-time decision-making. For this purpose, the system performance can be examined under different system configurations based on the dynamical effects of the risk factors in order to find out the best combination of operating policy of the system. Thus, the reconfiguration of the supply chain networks for different design alternatives can easily be achieved via PN based modelling.

To this end we first modelled the SC network using high level PN (see figure 2). Then a Petri net based simulation analysis is conducted for the performance evaluation of various scenarios and risk mitigation strategies under different levels of system parameters. In the simulation study, we generated a large number of customer order arrival events, and traced the order fulfillment process through the SC entities. The input data to the PN models is shown in Table 4.

Five independent replications for each mitigation actions were run for 120.000 operating minutes (250 days with one 8 hr shift) during the simulation of the PN model. In each run, the SC network is continuously loaded with customer-orders that are numbered on arrival, and each run produced one observation for each performance measure. Different random number seeds were used to prevent correlation between the parallel runs of the factorial experiment. By observing the mean flow time and average earliness/tardiness of the orders, it has been found that the system became stable after a warm-up period of 14.400 simulated minutes (30 days with one 8 hr shift). Thus, for each replication, the first 14.400 minutes was discarded to remove the effect of the start-up condition. After the warm-up period, performance measures of the entire supply chain are observed. The rules developed based on the mitigation scenarios are executed at the end of each 20.000 minutes to include the risk management process. According to the rule-based system, if it is necessary, the corresponding mitigation action is selected, then the related decisions are taken on the system parameters. Thus, dynamical effects of the different disruptive factors on the overall system performance and their interactions are observed in a real-time manner.

Table 5 summarizes the performance analysis results of the reference scenario and the scenario with the risk management strategies. The results presented here are the average values of all simulation run under different random seeds. However, different operational policies are simulated by the same set of random seeds to keep the SC environment the same for various scenarios.

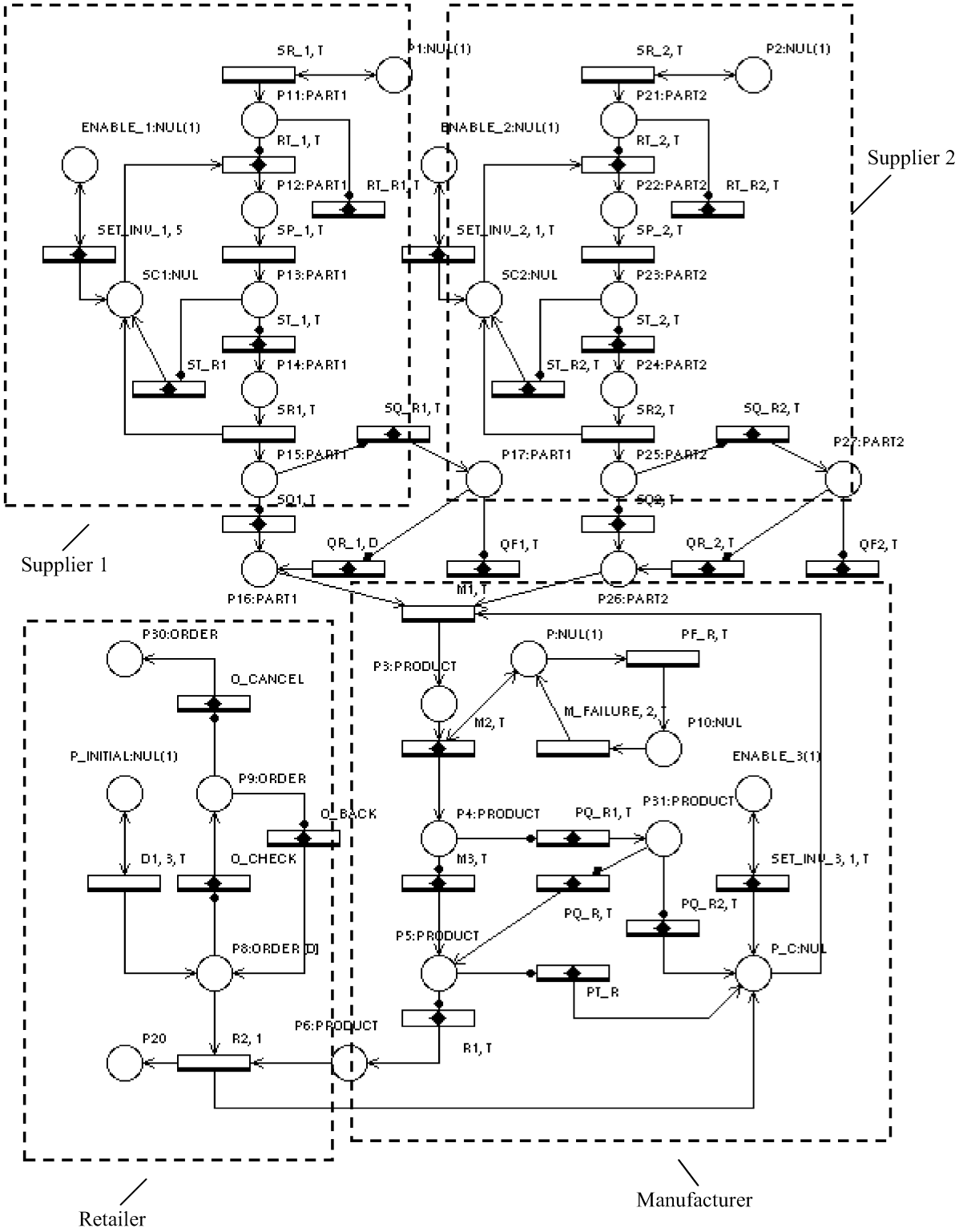


Figure 2: High-level Petri Net Model of the Supply Chain

Table 4: Simulation Parameter Settings

Parameter	**Value**
Customer order inter-arrival time	Exponential distribution with mean of 30 time units
Due time of the customer orders	Order arrival time + Exponential distribution with mean of 100 time units
Tardiness tolerance of orders	% 60 * (Due date – Arrival time)
Transportation time from supplier 1 to the manufacturer	Exponential distribution with mean 10 time units
Processing time by supplier 1	Uniform distribution range [22, 30]
Transportation time from supplier 2 to the manufacturer	Exponential distribution with mean 7 time units.
Processing time by supplier 2	Uniform distribution range [28, 35]
Reworking time for parts by suppliers	Exponential distribution with mean 15 time units.
Assembly operation time	Uniform distribution range [30, 35]
Reworking time for the assembled products	Exponential distribution with mean 10 time units.
Transportation time from manufacturer to the retailer	Exponential distribution with mean 15 time units.
Transportation failure risk	0.20
Quality failure risk for the parts / products	0.25
Quality failure risk for the reworked parts	0.40
System failure risk of manufacturer	Exponential distribution with mean 300 time units
Resolution time for system failure	Uniform distribution range [40, 60]
Order cancellation cost	50
Backlogging cost (additional cost for delayed orders)	5

Table 5: Performance Analysis Results of the Risk Management Scenarios

Performance measure	**The SC without RM**	**The SC with RM**
Total Cost	71,050	50,720
Customer order fill rate	% 65	% 83
Proportion of orders cancelled	% 35	% 17
Proportion of orders delayed	% 12	% 10
Number of orders delayed	432	372
Number of orders fulfilled	2322	2923
Number of orders cancelled	1229	603
Average tardiness of orders	6.14 time units	3.55 time units
Average earliness of orders	82.52 time units	86.86 time units

Compared to the reference case, we observe that the risk management decisions can increase the fill rate from % 65 to % 83, and decrease the cancellation rate from % 35 to % 17 with smaller total cost. Total cost refers the sum of the process cost and the cost incurred due to delayed deliveries, order cancellation, and related mitigation actions. The results show that disruptive factors and risk management actions have a large impact on the performance measures of the system. From the above results, a general conclusion is that the proposed rule-based system performs significantly better than the reference scenario without RM for all of the performance measures. With the help of the assessment and mitigation scenarios, the company is able to focus its activities and resources through the network more effectively. On the basis of the above modelling and performance evaluation of a SC, we can see that the proposed PN model well suits the modelling needs of SCs by providing a good visibility of material and order flow and a proper representation of risk factors in the network structure.

Since the proposed model has a flexible structure, it can easily be adapted to changing conditions by modifying only some parameters of the network, and employed to analyze the system performance. Considering the high implementation cost of SC networks, it is certainly worth choosing the best operating policy and system configuration by analyzing the developed system model, and adapting to changes over time. We note that in real industrial environments, the source of uncertainties is numerous and in order to get reliable results we need to have reliable estimation of these uncertainties. For instance, a variation in inter arrival times of the customer demand, operation times, supplier selection decisions, shipment size, production mix, inventory levels, and production and transportation failure rates can lead to different results and have different effects. Besides, the risks and their impact vary a lot according to the branch and company size (Hallikas et al. 2002). Therefore, instead of finding the optimal levels of all factors, which cover the ever-changing operation conditions, it is much more fruitful to develop a methodology that provides the procedures for the system designers, so that they can carry out the evaluation by themselves under their own operation conditions.

CONCLUSION

In this paper, a high-level PN based modeling methodology and a rule based approach for risk management in supply chains is presented. This study shows that PNs can be used effectively to model dynamic and stochastic nature of SC. They provide a thorough understanding of the control logic of the network structure, and can assist the evaluation of various operational strategies. The methodology presented here can be applied for designing, analyzing, specification, and evaluation of SCs as well as for solving uncertainty and risk related problems using a real-time decision making tool. By means of the modeling capabilities and formulating advantages, PNs can potentially play a significant role in risk modeling and analysis. Since real-world problems are large sized and complex, the future research issues still have to handle the large-scale problems.

REFERENCES

Chandra, C. and Kumar, S. 2000. "Supply Chain Management in Theory and Practice: a passing fad or a fundamental change?" *Industrial Management & Data Systems*, 100 (3), 100-116.

Chen, H.; Amodeo L.; Chu, F.; and Labadi, K. 2005. "Modeling and Performance Evaluation of Supply Chains Using Batch Deterministic and Stochastic Petri Nets." *IEEE Transactions on Automation, Science, and Engineering*, 2, 132-144.

Chopra, S. and Meindl, P. 2001, *Supply Chain Management: Strategy, Planning and Operation*, Prentice Hall, Upper Saddle River, NJ.

Desrochers, A.A. and Al-Jaar, R.Y. 1995. *Applications of Petri nets in Manufacturing Systems*. Institute of Electrical and Electronics Engineers Press, New York.

DiCesare, F.; Harhalakis, G.; Proth, J.M.; Silva, M. and Vernadat, F.B. 1993. *Practice of Petri Nets in Manufacturing*, Chapman & Hall, London.

Dong, M. and Chen, F.F. 2001. "Process Modelling and Analysis of Manufacturing Supply Chain Networks Using Object-Oriented Petri nets." *Robotics and Computer Integrated Manufacturing*, 17, 121-129.

Hallikas, J.; Virolainen, V.M.; and Tuominen, M. 2002. "Risk Analysis and Assessment in Network Environments: A dyadic case study." *International Journal of Production Economics*, 78, 42-55.

Jütter, U. 2005. "Supply Chain Risk Management." *International Journal of Logistics Management*, 16(1), 120-141.

Kersten, W.; Böger, M.; Hohrath, P.; and Spath H. 2006. "Supply Chain Risk Management: Development of a Theoretical and Empirical Framework." *Managing Risk in Supply Chains, (Edt. Kersten, W. and Blecker,T.)*, volume 1, 3-17, Erich Schmidt Verlag, Berlin.

Liu R.; Kumar, A.; and Van der Aalst, W. 2007. "A Formal Modeling Approach for Supply Chain Event Management." *Decision Support Systems* (*on-line first*).

Makajic-Nikolic, D.; Panic, B.; and Vujosevic, M. 2004. "Bullwhip Effect and Supply Chain Modelling and Analysis Using CPN tools." *Proc. of the Fifth Workshop and Tutorial on Practical Use of Coloured PNs and the CPN tools,* Aarhus, Denmark, October 8-11, DAIMI PB-570, 219-234.

Von Mevius, M. and Pibernik, R. 2004. "Process Management in Supply Chains - A new Petri net based approach." *In Proceedings of the 37th Hawaii International Conference on System Sciences,* IEEE 2004.

Moore, K.E. and Gupta, S.M. 1996. "Petri Net Models of Flexible and Automated Manufacturing Systems: a survey." *International Journal of Production Research*, 34 (11), 3001-3035.

Murata, T. (1989). "Petri Nets: Properties, Analysis and Applications." *In Proceedings of IEEE*, 77 (4), 541-580.

Mussigmann, N. 2006. "Mitigating Risk during Strategic Supply Network Modelling." *Managing Risk in Supply Chains (Edt. Kersten, W. and Blecker,T.)*, volume 1, 213-226, Erich Schmidt Verlag, Berlin 2006.

Nedeβ, C.; Friedewald, A.; Wagner, L.; and Neumann, L. 2006. "Risk Management in Maritime Transportation Networks." *Managing Risk in Supply Chains, (Edt. Kersten, W. and Blecker,T.)*, v. 1, 239-252, Erich Schmidt Verlag, Berlin.

RSoft Design Group, Inc. 2002. "Full spectrum photonic and network design automation." Retrieved November 29, 2005, from *http://www.rsoftdesign.com/products/network_modeling/Artifex/*

Viswanadham, N. and Raghavan, N.R.S. 2000. "Performance Analysis and Design of Supply Chains: A Petri net approach." *Journal of Operations Research Society*, 51 (10), 1158-1169.

Van der Vorst, J.G.A.J.; Beulens A.J.M.; and Van Beek, P. 2000. "Modelling and Simulating Multi-echelon Food Systems." *European Journal of Operational Research*, 122, 354-366.

Winkler, H. and Kaluza, B. 2006. "Integrated Performance – and Risk Management in Supply Chains – basics and methods." *Managing Risk in Supply Chains, (Edt. Kersten, W. and Blecker,T.)*, volume 1, 19-36, Erich Schmidt Verlag, Berlin.

Zhou, M.C. and DiCesare, F. 1993. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers, Norwell, Massachusetts.

Zurawski, R. and Zhou, M.C. 1994. "Petri Nets and Industrial Applications: A Tutorial." *IEEE Transactions on Industrial Electronics*, 41 (6), 567-582.

ACKNOWLEDGEMENT

This work is supported by the Scientific and Technological Research Council of Turkey (TUBITAK).

AUTHOR BIOGRAPHY

GONCA TUNCEL is a Post-doctoral fellow at the Laboratory G-SCOP in Grenoble, France. She received her PhD (2006) and MSc (1999) in Industrial Engineering from the Dokuz Eylul University (DEU) in Turkey. Her main research interests include discrete event systems, Petri Nets, modeling and analysis of production systems, dynamic scheduling, Flexible Manufacturing Systems (FMSs), logistics and SCM.

GÜLGÜN ALPAN is an Associate Professor of Industrial Engineering at the Laboratory G-SCOP in France. She received her PhD (1997) and MSc (1994) in Industrial Engineering from Rutgers University in New Brunswick, NJ, USA. Her main research interests are the modeling, analysis, and control of discrete event systems with applications in cyclic scheduling, supervisory control of FMS, logistics and SCM.

ANALYTICAL AND NUMERICAL MODELLING TECHNIQUES

PERFORMANCE OPTIMIZATION

TWO MONTE CARLO STUDIES FOR LATENT CLASS SEGMENTATION MODELS

Liberato Camilleri
Department of Statistics and Operations Research
University of Malta
Msida (MSD 06)
Malta
E-mail: liberato.camilleri@um.edu.mt

KEYWORDS
Information Criteria, Proportional Odds Model, Latent Class Model, Segmentation, EM algorithm

ABSTRACT

Model assessment and comparison are essential aspects of statistical inference. The likelihood ratio test is one of the main instruments for model selection; however, this is not appropriate when the model under consideration contains random effects. In this paper, we present two simulation studies for latent class segmentation models. The first Monte Carlo study compares the performance of seven Information Criteria in predicting the correct number of segments. The second study investigates factors that have an effect on segment membership and parameter recovery and affect computational effort.

1. INTRODUCTION – A GENERAL MODEL

Latent class models stands out as one of the major breakthroughs in market segmentation as they overcome the limitations of aggregate analysis and a-priori segmentation. In this approach, the segments and the model parameters within these segments are estimated simultaneously. Latent class methodology for market segmentation, suggested by (Green 2000) proposes the Proportional Odds model as a proper statistical model for ordinal data.

$$P\left(y_{jn}=r\left|\boldsymbol{\alpha},\boldsymbol{\beta}\right.\right)=F\left(\alpha_r+\mathbf{x}_j'\boldsymbol{\beta}\right)-F\left(\alpha_{r-1}+\mathbf{x}_j'\boldsymbol{\beta}\right) \qquad (1)$$

In this model y_{jn} is a rating response elicited by the n^{th} respondent for the j^{th} item; $\boldsymbol{\alpha}$ is a vector of threshold parameters; $\boldsymbol{\beta}$ is a vector of regression parameters and $\mathbf{x}_j$ are item or individual covariates. The choices of $F(.)$ considered are the Logistic, Normal and Extreme Value distributions respectively leading to the Logit, Probit and Complementary Log-Log links. For the segmentation model, the Proportional Odds model is extended by considering a Latent Class model with *K* segments.

$$P\left(y_{jn}=r\left|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\pi}\right.\right)=\sum_{k=1}^{K}\pi_k.P\left(y_{jn}=r\left|\boldsymbol{\alpha},\boldsymbol{\beta}_k\right.\right) \qquad (2)$$

where π_k is the proportion of respondents that are assigned to the k^{th} segment. The log-likelihood function is maximized through the EM algorithm. The merit of this model is that it allows for a probabilistic classification of respondents into segments and simultaneous estimation of a generalized linear regression model within each segment. When applying the above model to real data, the actual number of segments, *K*, is unknown and has to be specified. Unfortunately, the standard likelihood ratio statistic that tests between a *K*-segment model and a (*K*+1)-segment model does not have an asymptotic chi-square distribution and so it is not adequate to identify the appropriate number of segments in latent class models. Several information criteria have been proposed to identify this optimal number of segments.

2. INFORMATION CRITERIA TO IDENTIFY THE NUMBER OF SEGMENTS

Several Information Criteria have been proposed to compare Latent class models with different number of components (segments). These criteria stipulate different penalty terms to measure the complexity of the model. Most of the information criteria that are proposed are based on the bias-corrected log-likelihood given by:

$$C=-2\log L\left(\boldsymbol{\Psi}\right)+dc \qquad (3)$$

where *d* is the number of estimated parameters and *c* is a penalty constant. The second term, which is the penalty term, measures the complexity of the model. For instance the well-known Akaike information criterion (AIC), proposed by (Akaike 1974), arises when $c=2$. The major problem with the use of this criterion is that it relies on the same asymptotic properties as the likelihood ratio test. Many authors have observed that AIC tend to overestimate the correct number of segments. The Modified Akaike information criterion (MAIC) arises when $c=3$ and it penalizes complex models more

heavily than AIC. Another criterion that penalizes the log-likelihood more heavily is the Bayesian information criterion (BIC), proposed by (Schwarz 1978). For this criterion $c = \log(N)$, where N is the sample size. The penalty term of the BIC criterion depends on the sample size and favours models that are more parsimonious. For $N > 8$, BIC penalizes complex models more heavily than AIC and MAIC. BIC reduces the tendency of AIC and MAIC to fit too many segments.

The above criteria account for over-parameterization as more segments are derived. However, one must ensure that the segments are sufficiently separated for a particular solution. To examine the centroid separation between the segments, (Ramaswamy and Cohen 2000) use an entropy statistic to investigate the degree of separation in the estimated posterior probabilities.

$$E_s = 1 + \frac{EN(\hat{p}_{nk})}{N \log K} \tag{4}$$

where $\hat{p}_{nk}$ is the posterior probability that the n^{th} subject belongs to the k^{th} segment and $EN(\hat{p}_{nk})$ is the entropy given by $EN(\hat{p}_{nk}) = -\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{p}_{nk} \log(\hat{p}_{nk})$. $EN(\hat{p}_{nk})$ is the penalty term and penalizes models whose segments are poorly separated. When the segment centroids of these latent class models are well separated, $EN(\hat{p}_{nk})$ will be close to its minimum value of zero. If the segment centroids are not sufficiently separated for the number of segments specified then $EN(\hat{p}_{nk})$ will have a large value because the posterior probabilities for each observation are approximately equal. E_s is a relative measure bounded between 0 and 1. A value close to 1 indicates that the centroids of the derived segments are well separated. A value close to 0 indicates poor separation.

Another criterion that uses $EN(\hat{p}_{nk})$ to penalize a model for its complexity is the Classification Likelihood criterion (CLC) proposed by (Biernacki and Govaert 1997). This criterion minimizes

$$CLC = -2 \log L(\boldsymbol{\Psi}) + 2EN(\hat{p}_{nk}) \tag{5}$$

and the penalty on the log-likelihood depends on how well separated the fitted segments are. This criterion works best when the probabilities of segment membership happen to be similar. However, it tends to overestimate the correct number of segments when no restriction is placed on these probabilities.

Another criterion that uses the normalized form of $EN(\hat{p}_{nk})$ for choosing the number of segments is the Normalized Entropy Criterion (NEC) proposed by (Celeux and Soromenho 1996). This normalized form is given by:

$$NEC = \frac{EN(\hat{p}_{nk})}{\log L(\boldsymbol{\Psi}) - \log L(\boldsymbol{\Psi}^*)} \tag{6}$$

where $\log L(\boldsymbol{\Psi}^*)$ is the log-likelihood in the case of a single segment $(K = 1)$. This criterion has a shortcoming because $EN(\hat{p}_{nk}) = 0$ for $K = 1$ and so it is unable to decide between $K = 1$ and a value of K greater than one. (Biernacki, Celeux and Govaert 1999) proposed a modification to overcome this limitation. The modified criterion defines $NEC = 1$ for $K = 1$ and then chooses the number of segments to minimize NEC.

Another criterion, proposed by (Biernacki, Celeux and Govaert 1999), is the Integrated Classification Likelihood (ICL), which assumes that $N\pi_k$ are sufficiently large values. This criterion is chosen in an attempt to overcome the shortcomings of BIC and CLC. It minimizes

$$ICL = -2 \log L(\boldsymbol{\Psi}) + 2EN(\hat{p}_{nk}) + d \log(N) \tag{7}$$

3. STUDY DESIGN TO ASSESS INFORMATION CRITERIA PERFORMANCE

In order to assess the performance of the proposed criteria in identifying the correct number of segments, synthetic data sets were generated using a GLIM algorithm. The simulation was devised to mimic the application of (Camilleri and Green 2004) in which four car brands; four price values and two door features were generated to define the item attributes. In the application, a full profile approach was employed in which 32 items (cards) were generated where each card had a unique item attribute combination. This guaranteed a full factorial design. In the simulation, the item attribute values and the number of hypothetical respondents (N) were set the same as in the application. Two sets of uniformly distributed pseudo-random real values in the range [0,1] were used to generate the age and gender of each hypothetical subject. Pseudo random values less than 0.5 in the first set corresponded to male subjects. By transforming the pseudo random values in the second set using a linear relationship, the ages of the hypothetical subjects were generated to vary from 15 to 75 years.

To allocate the N hypothetical subjects into K segments, the proportions π_k were set the same as in the application. The cumulative probabilities, $q_0, q_1, ..., q_K$, were computed such that $q_k = \sum_{i=1}^{k} \pi_i$, where $q_0 = 0$ and $q_K = 1$. A set of uniformly distributed pseudo-random real values was then generated in the range [0, 1] to allocate the hypothetical subjects to one of the K segments. Subjects whose corresponding pseudo-random values were in the range $[q_{k-1}, q_k]$ were allocated to the

k^{th} segment. This classification gave each subject a random segment allocation.

To simulate the subjects' utility responses, the utility model (linear predictor) and the parameter values for the K segments were set the same as in the application. The utility model included both main effects and interaction terms of the item and individual covariates. 32 synthetic data values or utility values were generated for each hypothetical subject by substituting the parameter values and the values of the item and individual covariates in the utility model.

Error terms ε_i were added to these utility values to have either a logistic or a normal or an extreme value distribution. These error terms were generated by transforming pseudo-random real values u_i in the range [0,1] from a uniform distribution. If ε_i has a logistic distribution then $\varepsilon_i = \ln\left[u_i/(1-u_i)\right]$; $\varepsilon_i = \Phi^{-1}(u_i)$ if ε_i have a normal distribution and $\varepsilon_i = \ln\left[-\ln(1-u_i)\right]$ if ε_i have an Extreme value distribution. A set of six specified cut-point values α_r was used to convert these modified utility values to rates ranging from 1 to 7. Items (cards) whose modified worth values were in the range (α_{r-1}, α_r) were rated in the r^{th} worth category. This classification gives the rating responses of each hypothetical subject a random category allocation.

4. RESULTS OF THE FIRST STUDY

An empirical comparison was carried out to determine which of the above criteria best select the correct number of segments in a Latent Class model. The study compared the performance of the more recently suggested criteria such as CLC, NEC and ICL with classical procedures such as AIC, MAIC and BIC. By assuming a Logistic distribution, fifteen data sets were generated using the same utility model, design matrix and parameter values. These data sets were simulated using $N = 310$ and $K = 4$. Each simulated data set was re-fitted four times varying the number of segments from three to six clusters. The log-likelihood and the entropy were recorded to determine the number of segments that minimize the specified criteria. Solutions that were spurious were eliminated and a different random start was considered to initialize the EM algorithm. The numbers of parameters for the latent class model with 3, 4, 5 and 6 segments were respectively d = 58, 76, 94 and 112. These include the parameters of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$.

It is evident from tables 1 and 3 that the reduction in the log-likelihood is significantly larger when fitting 3 and 4 segments when compared to fitting 4, 5 and 6 segments. This implies that the reduction in the log-likelihood tends to become smaller when the number of fitted segments exceeds the number of true segments.

	K=3	K=4	K=5	K=6
Set	$-2\log L(\hat{\Psi})$	$-2\log L(\hat{\Psi})$	$-2\log L(\hat{\Psi})$	$-2\log L(\hat{\Psi})$
1	15059.3	14083.8	14062.4	13933.5
2	14987.7	14215.6	14099.2	14090.3
3	15018.0	14106.8	14039.9	13889.7
4	15107.5	14195.2	14126.9	14047.3
5	15006.8	14177.2	14107.8	14034.9
6	15130.1	14191.1	14141.6	14115.3
7	14972.7	14143.8	14096.5	14025.8
8	15077.6	14148.0	14082.7	14043.9
9	15160.7	14096.1	14062.8	13975.8
10	15032.9	14059.5	13949.6	13822.5
11	14857.4	14018.3	13943.7	13825.2
12	14984.5	14170.1	14125.6	14054.2
13	14931.4	14186.0	14155.4	14087.4
14	14995.5	14154.7	14098.2	14021.8
15	15206.0	14363.9	14296.6	14184.3

Table 1: Deviances for 3, 4, 5 and 6 segments

	K=3	K=4	K=5	K=6
Set	**Entropy**	**Entropy**	**Entropy**	**Entropy**
1	45.54	9.036	16.23	25.89
2	17.89	10.43	19.93	13.40
3	42.88	8.590	18.93	28.48
4	44.91	10.40	26.44	44.34
5	48.86	12.15	15.82	40.77
6	81.86	11.72	23.59	34.77
7	12.21	10.04	20.50	18.60
8	71.10	10.01	10.76	33.83
9	53.89	10.75	18.65	24.11
10	73.70	15.25	16.70	26.16
11	43.72	10.06	21.99	23.57
12	74.00	11.15	9.910	20.94
13	76.76	15.98	16.36	35.73
14	70.83	10.99	26.35	25.13
15	8.210	9.998	20.45	43.42

Table 2: Entropies for 3, 4, 5 and 6 segments

		$-2\log L(\hat{\Psi})$	Entropy
***K*=3**	Mean	15035.2	51.09
	Standard Deviation	90.77	24.04
	Minimum	14857.4	8.21
***K*=4**	Mean	14154.0	11.10
	Standard Deviation	80.51	2.049
	Minimum	14018.3	8.59
***K*=5**	Mean	14092.6	18.84
	Standard Deviation	83.60	4.818
	Minimum	13943.7	9.91
***K*=6**	Mean	14016.8	29.02
	Standard Deviation	98.36	9.202
	Minimum	13822.5	13.4

Table 3: Descriptive statistics for deviances and entropies

Table 3 displays another interesting result. The mean and standard deviation of the entropy are smallest when 4 segments are fitted. These two measures increase when the number of fitted segments exceeds the number of true segments. The addition of extra segments increases the mean entropy because the centroid separation between the segments is reduced. The number of hypothetical subjects in these extra segments also affects the size of the entropy.

Tables 4 and 5 show segment membership recovery of the 310 hypothetical subjects for the 4th and 7th data sets. The entropies for these two data sets were respectively 44.34 and 18.60. This implies that the entropy increases as the number of hypothetical subjects in these extra segments increases. When the number of fitted segments exceeds the number of true segments there are two possible outcomes. If four of the fitted segments include a large proportion of the hypothetical subjects such that the other two segments have small frequencies then the segments are more likely to be well separated and the posterior probabilities tend to be close to either 0 or 1. This yields a small entropy value. However, if one or more of the larger segments split, locations of the new clusters are relatively close. Moreover, the proportion of correctly classified hypothetical subjects decreases. So the posterior probabilities are more likely to be distant from either 0 or 1, yielding a larger entropy value. This explains why the dispersion of the entropies increases when too many segments are fitted.

		Fitted Segments					
		1	2	3	4	5	6
True Segments	1	52	0	0	0	36	0
	2	0	67	0	0	0	0
	3	1	0	45	0	1	31
	4	0	0	0	77	0	0

Table 4: Segment allocations for the 4th data set

		Fitted Segments					
		1	2	3	4	5	6
True Segments	1	97	0	0	0	0	0
	2	0	58	0	0	0	3
	3	1	0	56	0	15	0
	4	0	0	0	74	0	6

Table 5: Segment allocations for the 7th data set

Table 6 shows the result of the empirical comparison between 7 different criteria to determine which one best selects the correct number of segments in a Latent Class model. It is evident that the modern procedures NEC and ICL outperform the classical procedures AIC and MAIC. AIC and MAIC have a tendency to fit too many segments, whereas BIC penalize complex models more heavily than AIC and MAIC. The penalty term of most criteria depends on one or more of the following features; number of estimated parameters, number of subjects and entropy. Criteria that combine two or more of these features in the penalty term are superior to those that contain only one feature. The penalty term of AIC and MAIC depends solely on number of estimated parameters and the penalty term of CLC depends entirely on the entropy. All three are inferior to the other criteria in recovering the true number of segments.

K	AIC	MAIC	BIC	CLC	ICL	NEC	E_s
3	0	0	0	0	0	0	0
4	0	1	12	0	14	14	11
5	2	2	1	1	1	0	4
6	13	12	2	14	0	1	0

Table 6: Criteria Performance (Logistic distribution)

A further task was included in the study to investigate how the choice of the distribution function affects the performance of the Information Criteria in selecting the optimal number of clusters and how it affects segment membership recovery. To carry out this task, a further thirty data sets were generated using the same utility model, design matrix and parameter values as in the previous assignment. A Normal distribution was assumed to generate the first fifteen data sets and an Extreme value distribution was assumed to generate the rest. These data sets were simulated using $N = 310$ and $K = 4$. Each simulated data set was again re-fitted four times varying the number of segments from three to six clusters. The log-likelihood and the entropy were recorded to determine the number of segments that minimize the specified criteria.

Correct segment allocation deteriorates slightly when an Extreme value distribution is assumed; however, segment-membership recovery improves when using a Normal distribution. Tables 7 and 8 show the results of the empirical comparison of the Information Criteria when a Normal or an Extreme Value distribution was assumed. Both tables display that the procedures ICL, NEC and BIC outperform the procedures AIC, MAIC and CLC. This implies that the performance of the Information Criteria in selecting the correct number of segments is not affected much by the choice of the distribution.

K	AIC	MAIC	BIC	CLC	ICL	NEC	E_s
3	0	0	0	0	0	0	0
4	0	1	14	1	15	15	12
5	4	5	1	2	0	0	3
6	11	10	0	12	0	0	0

Table 7: Criteria Performance (Normal distribution)

K	AIC	MAIC	BIC	CLC	ICL	NEC	E_s
3	0	0	0	0	0	0	0
4	0	0	10	0	13	12	10
5	0	0	4	0	2	1	4
6	15	15	1	15	0	2	1

Table 8: Criteria Performance (Extreme value distribution)

5. FACTORS AFFECTING THE PERFORMANCE OF LATENT CLASS MODELS

A further task was to examine the performance of latent class models by modifying a number of factors. Three of the factors that are highlighted in literature (Vriens, Wedel and Wilms 1996; Wedel and DeSarbo 1995) as having potential effect on model performance include:

- Number of simulated respondents
- Number of segments
- Distribution of the dependent variable

The above three factors reflect a variation in conditions in many practical applications and which are expected to affect the performance of the model fit. The following six measures are normally used to assess computational effort, parameter recovery, predictive power, goodness of fit and segment membership recovery.

- The percentage of variance, R^2 accounted for by the latent class model is a measure of the goodness of fit.

- The number of iterations required for convergence is a measure of the computational effort.

- The root-mean-squared error between the true and estimated parameters and the root-mean-squared error between the true and estimated segment membership probabilities are measures of parameter recovery.

$$RMS\left(\hat{\boldsymbol{\beta}}\right) = \left[\sum_{p=1}^{P} \frac{\left(\beta_p - \hat{\beta}_p\right)^2}{P}\right]^{\frac{1}{2}} \tag{8}$$

  $\hat{\beta}_p$ and β_p are respectively the estimated and true parameters; whereas P is the number of parameters.

$$RMS\left(\hat{\boldsymbol{\pi}}\right) = \left[\sum_{k=1}^{K} \frac{\left(\pi_k - \hat{\pi}_k\right)^2}{K}\right]^{\frac{1}{2}} \tag{9}$$

  $\hat{\pi}_k$ and π_k are respectively the estimated and true segment membership probabilities; whereas K is the number of segments.

- The root-mean-squared-error between the true and predicted responses is a measure of the predictive power.

$$RMS\left(\hat{y}\right) = \left[\sum_{n=1}^{N}\sum_{j=1}^{J} \frac{\left(y_{nj} - \hat{y}_{nj}\right)^2}{N.J}\right]^{\frac{1}{2}} \tag{10}$$

  $\hat{y}_{nj}$ and y_{nj} are respectively the estimated and true responses; whereas N and J are respectively the number of simulated respondents and the number of items (cards) assessed by each subject.

- The percentage number of subjects that are correctly classified into their true segments is a measure of segment membership recovery. A subject is assigned to the segment with highest posterior probability.

6. STUDY DESIGN TO ASSESS THE FACTORS THAT AFFECT MODEL PERFORMANCE

In order to assess the factors that affect the performance of latent class models, synthetic data sets were generated using a GLIM algorithm. The simulation was devised to mimic the application of (Camilleri and Green 2004). The design and the utility model (linear predictor) were set the same as in the application. The age and gender of the N hypothetical subjects and their segment allocation were generated using a similar procedure described in the first simulation study. To simulate subjects' rating responses, a set of parameters was specified such that all the main effects and interaction terms in each segment were assigned a parameter. 32 synthetic response values were generated for each hypothetical subject by substituting the parameter values and the values of the item and individual covariates in the utility model.

The number of simulated respondents was varied at two levels: 200 and 310. These levels represent a reasonable range of sample sizes that are reported in several segmentation applications (Wittink, Vriens and Burhenne 1994; Wedel and Steenkamp 1991). Standard theory on statistical inference suggests that a greater number of simulated respondents improve the precision of the estimated segment-level parameters.

The number of segments was also varied at two levels. A two-segment and a four-segment condition were used because these represent the range of segments commonly found in segmentation applications (Wedel and Steenkamp 1989; DeSarbo, Oliver and Rangaswamy 1989). It is expected that a greater number of segments deteriorate the precision of the estimated segment-level coefficients as a greater number of model parameters have to be estimated.

The Proportional Odds model can accommodate three possible distribution functions. The Logistic, Normal and Extreme value distributions, which respectively lead to logit, probit and complementary log-log link functions, were all considered.

A problem associated with the application of the EM algorithm to latent class models is its convergence to local maxima. It is caused by the likelihood being multimodal, so that the algorithm becomes sensitive to the starting values used. The problem of convergence to local optima becomes more conspicuous when the component densities are not well separated and when the number of estimated parameters is large. This will lead to a relatively weak update in the E-step (Wedel and Kamakura 2000). To overcome this problem five starting values were considered for each combination of the factor levels defined above. These were selected from a wide range of seed numbers. Another problem with the EM algorithm is that the fitted segments are very often a swapped version of the true segments. To overcome this problem, the parameters were chosen to contrast considerably between segments. The purpose was to simplify the identification of the correct correspondence between the fitted segments and the true segments.

7. RESULTS OF THE SECOND STUDY

Five data sets were generated for each factor level combination according to the type of distribution, number of subjects and number of segments. Each simulated data set was re-fitted using a latent class model. Solutions that were considered spurious were eliminated and a different random start was considered to initialize the EM algorithm. The statistics $RMS(\hat{\boldsymbol{\beta}})$ $RMS(\hat{\boldsymbol{\pi}})$ and $RMS(\hat{y})$ were computed after permuting the parameters and predicted responses to match estimated and true segments optimally. All the six measures were averaged over these five data sets.

Table 9 exhibits some differences in the R^2 measures between the types of distributions. The goodness of fit is improved when the choice of the distribution is Normal or Logistic. The value of R^2 increases with an increasing number of segments and a decreasing number of subjects.

Distribution	Number of respondents	Number of segments	R^2
Logistic	200	2	0.8516
Normal			0.8612
Extreme			0.8394
Logistic	310		0.8455
Normal			0.8563
Extreme			0.8222
Logistic	200	4	0.9217
Normal			0.9366
Extreme			0.8876
Logistic	310		0.9158
Normal			0.9193
Extreme			0.8685

Table 9: Measures of goodness of fit

Table 10 demonstrates that the number of segments mostly affects computational effort. An increase in the number of segments increases the number of iterations required. The number of simulated respondents and the choice of the error distribution have negligible effect on computational effort.

Distribution	Number of respondents	Number of segments	Number of iterations
Logistic	200	2	29.2
Normal			30.4
Extreme			31.6
Logistic	310		32.2
Normal			28.8
Extreme			29.4
Logistic	200	4	36.6
Normal			37.8
Extreme			35.8
Logistic	310		38.2
Normal			37.6
Extreme			38.2

Table 10: Measures of computational effort

Table 11 exhibits that the number of segments and the choice of distribution affect the percentage of correctly classified subjects; however, the number of subjects has negligible effect on segment-membership recovery. Segment-membership recovery deteriorates slightly with an increase in the number of segments and this deterioration worsens when an Extreme value distribution is used. When hypothetical subjects are allocated to segments, a mismatch in a four-segment solution is more likely to occur than in a two-segment solution. Segment-membership is recovered best when using a Normal or a Logistic distribution.

Distribution	Number of respondents	Number of segments	Segment recovery
Logistic	200	2	98.8%
Normal			99.0%
Extreme			95.1%
Logistic	310		99.3%
Normal			99.4%
Extreme			95.2%
Logistic	200	4	98.5%
Normal			98.5%
Extreme			94.3%
Logistic	310		98.7%
Normal			98.9%
Extreme			94.0%

Table 11: Measures of segment membership recovery

Table 12 exhibits that the mean $RMS(\hat{\boldsymbol{\pi}})$ is not affected by the choice of distribution used. The mean $RMS(\hat{\boldsymbol{\pi}})$ decreases with an increase in the number of segments and hypothetical subjects. Increasing the number of model parameters and increasing the sample size improve the probabilities of segment membership.

Distribution	Number of respondents	Number of segments	$\mathbf{rms}(\hat{\boldsymbol{\pi}})$
Logistic	200	2	0.0429
Normal			0.0461
Extreme			0.0433
Logistic	310		0.0325
Normal			0.0313
Extreme			0.0329
Logistic	200	4	0.0238
Normal			0.0256
Extreme			0.0219
Logistic	310		0.0197
Normal			0.0195
Extreme			0.0199

Table 12: Measures of segment proportion recovery

Table 13 demonstrates that the type of distribution affects parameter recovery. The mean $RMS(\hat{\boldsymbol{\beta}})$ is lowest when the choice of the error distribution is Normal and highest when the Extreme value distribution is used. Parameter recovery improves with an increase in the number of simulated subjects but deteriorates with an increase in the number of segments.

Distribution	Number of respondents	Number of segments	$\text{rms}(\hat{\beta})$
Logistic	200	2	0.2215
Normal			0.2109
Extreme			0.2678
Logistic	310		0.1633
Normal			0.1527
Extreme			0.2134
Logistic	200	4	0.2596
Normal			0.2578
Extreme			0.3073
Logistic	310		0.2236
Normal			0.2227
Extreme			0.2511

Table 13: Measures of parameter recovery

Table 14 shows that an increase in the number of subjects and a decrease in the number of segments improve the predictive accuracy. The Extreme-value distribution yields the highest mean value of $RMS(\hat{y})$ implying that the predictive accuracy deteriorates when this distribution is used.

Distribution	Number of respondents	Number of segments	$\text{rms}(\hat{y})$
Logistic	200	2	2.1629
Normal			2.1538
Extreme			2.6664
Logistic	310		2.1215
Normal			2.1036
Extreme			2.5319
Logistic	200	4	3.0516
Normal			3.0429
Extreme			3.1219
Logistic	310		3.0217
Normal			3.0424
Extreme			3.1056

Table 14: Measures of predictive power

8 CONCLUSIONS

In the first Monte Carlo study, several Information Criteria were proposed to determine the optimal number of segments in a Latent Class model. These criteria specify different penalty terms to measure the complexity of the model. These penalty terms depend on one or more model features, which include the number of estimated parameters, number of hypothetical subjects and entropy. This study illustrates that information Criteria that combine two or more model features in the penalty term outperform those criteria that include solely one feature. This study also demonstrates that the performance of these Criteria is not affected by the choice of the distribution function.

The second simulation study reveals several appealing results. Goodness of fit improves and computational effort increases with a larger number of segments; however, parameter recovery, segment membership recovery and predictive accuracy improve with a smaller number of segments. These results conform to standard theory on statistical inference. The choice of the error distribution has noticeable effect on model performance. The Normal and Logistic distributions outperform the Extreme value distribution. These two distributions yield better fits and improve predictive power, segment membership recovery and parameter recovery. In general, the effects of the number of simulated subjects on parameter recovery, segment membership recovery, computational effort, predictive accuracy and goodness of fit are smaller than the effects of the number of segments and distribution choice.

ACKNOWLEDGEMENTS

The author gratefully acknowledges Dr Michael Green for his contribution in developing a simulation algorithm that generates synthetic data for Latent Class model fitting.

REFERENCES

Aitkin, M., Anderson, D., Francis, B. and Hinde, J. (1994), *Statistical Modelling in GLIM*, Oxford Science Publications.

Akaike, H. (1974), A New Look at Statistical Model Identification, *IEEE Transactions on Automatic Control*, AC-19, 716-723.

Biernacki, C., Celeux, G. and Govaert, G. (1999), An Improvement of NEC Criterion for assessing the number of Clusters in a Mixture Model, *Pattern Recognition Letters*, 20, 267-272.

Biernacki, C. and Govaert, G. (1997), Using the Classification Likelihood to Choose the Number of Clusters, *Computing Science and Statistics*, 29, 451-457.

Camilleri, L. and Green, M. (2004), Statistical Models for Market Segmentation, *Proceedings of the 19th International Workshop Statistical Modelling, Florence*. 120-124.

Camilleri, L. and Portelli, M. (2007), Segmenting the heterogeneity of tourist preferences using a Latent Class model combined with the EM algorithm, *Proceedings of the 6th APLIMAT International Conference, Bratislava*. 343-356.

Celeux, G. and Soromenho, G. (1996), An Entropy Criterion for assessing the number of clusters in a Mixture model, *Classification Journal*, 13, 195-212.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977), Maximum Likelihood from Incomplete Data via the EM algorithm, *Journal of the Royal Statistical Society*, B, 39, 1-38.

DeSarbo, W.S., Oliver, R.L. and Rangaswamy, A. (1989), A Simulated Annealing Methodology for Clusterwise Linear Regression, *Psychometrika*, 54, 707-736.

Francis, B., Green, M. and Payne, C. (1993), *The GLIM 4 manual*, Oxford Science Publications.

Green, M. (2000), Statistical Models for Conjoint Analysis, *Proceedings of the 15th International Workshop on Statistical Modelling, Bilbao*. 216-222.

Nelder, J.A and Wedderburn, R.W.M. (1972), Generalized Linear Models, *Journal of the Royal Statistical Society*, A, 135, 370-384.

Ogawa, K. (1987), An Approach to Simultaneous Estimation and Segmentation in Conjoint Analysis, *Marketing Science*, 6, 66-81.

Ramaswamy, V. and Cohen, S.H. (2000), Latent Class Models for Conjoint Analysis, in Gustafsson, A., Hermann, A. and Huber, F., ed., *Conjoint Measurement: Methods and Applications*, Springer, 415-446.

Schwarz, G. (1978), Estimating the Dimension of a Model, *The Annals of Statistics*, 6, 461-464.

Vermunt, J.K. (2004), An EM algorithm for the estimation of parametric and non-parametric hierarchical non-linear models, *Statistica Neerlandica*, 58, 220-233.

Vriens, M., Wedel, M. and Wilms, T. (1996), Conjoint Segmentation Methods A Monte Carlo Comparison, *Journal of Marketing Research*, 23, 73-85.

Wedel, M. and DeSarbo, W.S. (1995), A Mixture Likelihood Approach for Generalized Linear Models, *Journal of Classification*, 12, 1-35.

Wedel, M. and Kamakura, W.A. (2000), *Market Segmentation: Conceptual and Methodological Foundations*, Kluwer Academic Publishers.

Wedel, M. and Kistemaker, C. (1989), Consumer Benefit Segmentation using Clusterwise Linear Regression, *International Journal of Research Market*, 6, 45-49.

Wedel, M. and Steenkamp, J.B. (1989), Fuzzy Clusterwise Regression Approach to Benefit Segmentation, *International Journal of Research in Marketing*, 6, 241-258.

Wedel, M. and Steenkamp, J.B. (1991), A Clusterwise Regression Method for Simultaneous Fuzzy Market Structuring and Benefit Segmentation, *Journal of Market Research*, 28, 385-396.

Wittink, D.R., Vriens, M. and Burhenne, W. (1994), Commercial Use of Conjoint Analysis in Europe, Results and Critical Reflections, *International Journal of Research in Marketing*, 11, 41-52.

AUTHOR BIOGRAPHY

LIBERATO CAMILLERI studied Mathematics and Statistics at the University of Malta. He received his PhD degree in Applied Statistics in 2005 from Lancaster University. His research specialization areas are related to statistical models, which include Generalized Linear models, Latent Class models, Multi-Level models and Random Coefficient models. He is presently a lecturer in the Statistics department at the University of Malta.

SIMULATION FOR OPTIMIZING THE GENERATION OF ELECTRIC ENERGY CONSIDERING RENEWABLE SOURCES

Eugeniusz M. Sroczan
Institute of Electric Power Engineering
Poznan University of Technology
ul. Piotrowo 3A, 60-965 Poznan, Poland
E-mail: sroczan@put.poznan.pl

KEYWORDS

Energy, interactive programs, interactive simulation, optimization, natural resources.

ABSTRACT

Optimal power load of committing power plants allows the generated energy to get the minimum cost. It results from the technological structure of power units' set installed in the examined power system (PS) and committed to meet the power demand. The special structure of PS is important in case of failure or inconvenient weather conditions, for example too low or too high wind velocity for wind power plants (WPP) or not enough rain causing lack of flows in the hydro plant. The proposed simulator consists of some procedures that enable considering real values of energy prices and variable magnitude of energy demand caused by energy consumers. The optimal contribution of committing power plants to cover the power system load is obtained by using fuzzy linear programming.

INTRODUCTION

Simulation for the impact of defined groups of power plants on the power system is needed to investigate the capability of regulating power control at different load and conditions of electric energy production. This problem is particularly important with regard to long-term stability of enlargement and modification or reengineering of power system. It is possible to describe the optimal structure of the system with optimal participation of the group of power plants converting the renewable energy. It is necessary to take into account the possibility of change the committing power units' set structure. So far it has been balancing the electric power load in order to be ready for demand.

This kind of simulation is also essential for the case of necessary change of aggregate in committing units. The reason for this change may be the administered renewable energy or it may result from an equipment malfunction or insufficient quality and volume of primary energy (insufficient speed of wind or lack of water inflow).

In large scale wind farms there are introduced some deviations in the planned power generation due to rapid changes of wind that may lead to power system operation problems.

The described simulator concerns the two basic renewable sources: wind power plants (WPP) and hydropower plants (HPP). The combined heat and power stations and power stations fired with coal (also with the participation of biomass co-burning) are treated as conventional power plants.

The simulated structure of changes within the power unit set causes essential (critical) effects in the processes of exploitation and considers the development of electrical power system – in the area of power plants and electrical grids (Batierra at al. 1998, Kalinowski at al. 1999, Malko 1997, Sroczan 1996). Some simulation routines for teaching and training the power system managers has been described in the research work (Sroczan 2006).

Mode of operation of the sources load is based on the cost minimization of generated energy using the developed simulator. Structure of the simulator (Sroczan 2006) consists of procedures simulating the policy of generation and checking the economic quality of the balance between power generation and power demand in the considered PS.

The essential problems of Local Energy Market (LEM) are concerned to balance:

- energy production in accordance with consumers' demands and with regard to the technical and economic limitations as well;
- power flow in the local power system, considering a varied set of energy sources, especially renewable energy sources.

The main aim of this paper is to simulate the effect of power plants' structure (with sources defined as a kind of primary energy) with intention of generated electric energy costs' minimization. Additional limiting conditions are variable weather and different users' behaviour towards electric energy use, defined as a profile of energy user.

The proposed attempt is based on the developed algorithm used to calculate the real costs of energy, taking into consideration the stated circumstances of demand and possibilities of loading the defined set of renewable energy sources. In the presented paper there are the results of research done for the given set of following types of power plants: coal fired, hydropower (HPP), pumped storage (PSHP) and wind power (WPP) (Sroczan 2006). The cost of generated energy should be calculated for the defined period of optimization (horizon of planning) and for diverse types and sets of power plants. Finally, it should define the rules of optimal policy for development and maintenance of the generation structure for the given LEM.

CALCULATION FOR POWER LOAD OF RENEWABLE ENERGY SOURCES

Change of the electric power demanded by energy end-users causes the changes of energy generation, transmission and distribution cost (Sroczan and Urbaniak 2004, Kalinowski at al. 1999). The cost of transmission and distribution are negligible in the discussed PS with regard to constant configuration of the network.

The PS operator defines the assumed value of the load covered by ecological sources. In the attempt presented in (Sroczan 2006) it is the RER – renewable energy ratio, defined as contribution of HPP and WPP to the PS generation process:

$$RER = \frac{A_{RE}}{A_T + A_{RE}} \tag{1}$$

where: A_T – volume of energy generated by thermal units, A_{RE} – volume of energy generated by HPP and WPP.

The RER value is diverse because of changes of factors γ and κ converting real value of energy produced by hydropower and wind-power plants (Sroczan 2006) to the comparable level of generation cost in coal-fired power plants. Total cost of generation C_t, as a sum of cost of each plant $C_i(P_{gi})$ is minimized:

$$C_t \rightarrow \min\left\{\sum_{i=1}^{n} C_i\left(P_{gi}\right)\right\} \tag{2}$$

The calculated values of economical effectiveness for the examined set of n committing units are obtained using the formula (2) and methods of discounted cash flows in order to find out if the investment is profitable or not.

Usually, the methods are total cost of ownership (TCO) or net present value (NPV) and internal rate of return (IRR). They are applied to compare the considered alternatives. Particularly, the cost was calculated using the NPV method as follows:

$$NPV = C_{Fo} + \sum_{t=1}^{N} \frac{\alpha \cdot C_{Ft}}{(1+r)^t} \tag{3}$$

where: C_{Fo} – capital investments' cost, C_{Ft} – cash flow in the t^{th}-year, α- risk factor, r – discount rate, N –lifetime of power plant, t – year of exploitation.

Therefore, the developed simulator can support the decisions of PS manager in the area of priority of loading the committing power plants.

STRUCTURE OF SIMULATOR

The process of choosing the optimal structure for the sources in the given PS and for the period of time is based on procedures of simulation implemented in the developed simulator (fig. 1.).

The developed simulator should support the decision of choosing the proper policy for extension. Moreover, it should help to realize the policy of diverse modification of the power plants' structure in the given PS.

Computational objectives appearing in PS, which include the power plants with limited energy production – WPP, pumped storage hydropower plants PSHP and HPP, consider acting according to following data and parameters:

- assumed structure of energy generation defined as value of RER factor;
- energetic characteristics of committed power plants (or units);
- electric power demand in the discussed period of time.

Limitation and constraints of energy generation refers especially to ecological energy, depending (except the PS and consumer effects) additionally on wind and hydro resources.

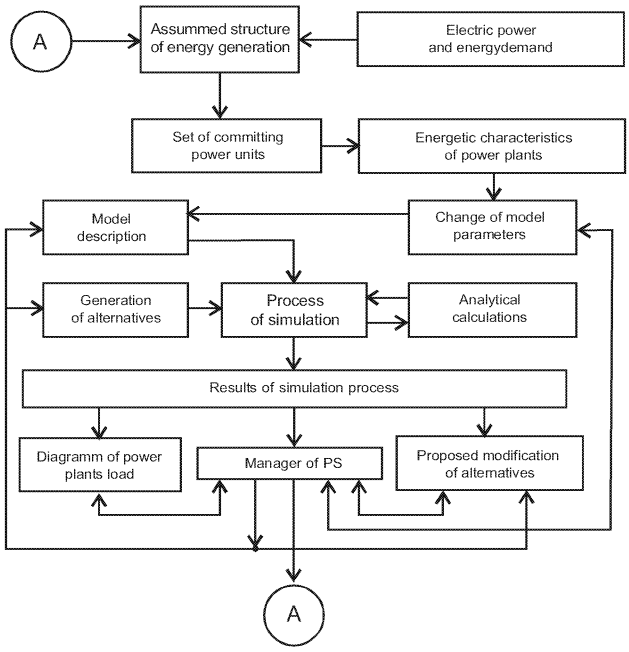


Figure 1: Structure of the simulator destined for optimizing the structure of PS generation

The costs of energy generation are calculated with regard to energetic characteristics of committing power units in the mode described in paper (Sroczan 1996).

SIMULATED COST OF ENERGY

Assuming the possible values of wind speed (annual average speed) in the given wind farms and rated power of units it is possible to estimate the values of energy prices. For the wind and hydro generators the price of energy includes: T_r – annual time of load with the rated power P_{WPr} r - discount rate, capital cost of investment and cost of maintenance and operation.

Priority of load the committing power plants is calculated by using the relations (1), (2) and (3), results are shown in the figures which illustrate the power load of plants (fig. 2) and the increase of generation cost versus the demand power increment ΔP_g of PS (fig. 3).

There are the example results shown in Fig. 3. They illustrate the prices at the distribution level, without the transmission and distribution cost.

The final price of the electric energy for a delivery company is calculated as an average value, defined in the contract between power plant and wholesale energy pro-

vider for the given generated power and time range.

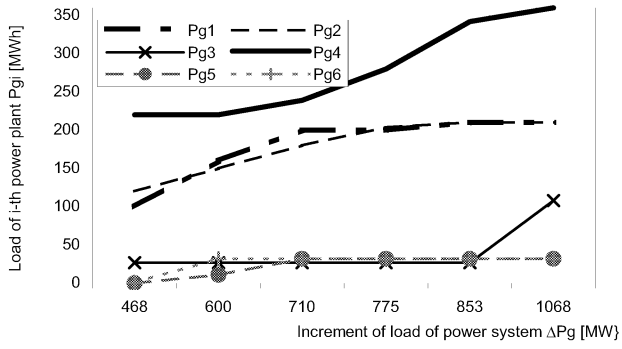


Figure 2: The load of committing power plant in accordance with PS power load increment

The expected structure of energy generation is described as result of optimization process solved by using FLP:

$$F_{SC} = \sum_t \left\{ \sum_j \left[\Delta P_j c_j \right] + \sum_i c_i \left[H(t), W(t), S(t) \right] \right\} \Delta t \qquad (4)$$

$$C^T \cdot x \preceq F_0 \qquad (5)$$

$$A \cdot x \preceq g \qquad (6)$$

where: F_{SC}, F_0 – real and optimal cost of energy generation; g – constraints caused by limitations: H,W - hydropower and wind plants, S – state of power system; T, Δt - time and time increment of the optimization period.

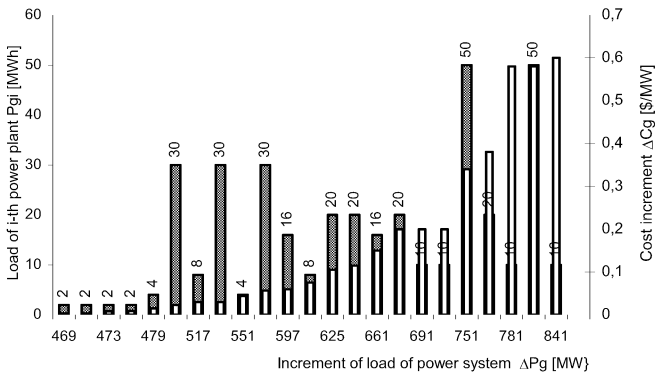


Figure 3: The results of simulation for the energy generation cost

RESULTS OF SIMULATION

The relation between the increasing cost of generation in thermal plants and values of γ and κ coefficients (Sroczan 1996, 2006) affects the structure and the load value of the subset of HPP and WPP as well as cost of generation *Ct*. The optimal case for renewable sources is obtained when the increase of costs for both types of renewable sources of energy (sets of HPP and WPP) are equal.

An increase in RER ratio affects an average cost of generated and consumed energy. For example, in loaded set of units, when RER increases to the range of 50 % the decreasing cost of generated energy is approximately 12 %.

CONCLUSIONS

The developed simulator runs in the interactive mode. The effect of PS sources structure on generation costs is solved using some procedures of simulation. They enable calculation of the expected costs of energy for the given wholesale providers and power plants, assuming the varied structure of sources – types of power plant.

The stated volume of "green energy" is defined by PS operator, using the RER factor.

The structure of PS is important in case of failure or inconvenient weather conditions, for example too low or too high wind velocity for WPP or weak rain and lack of flows in hydro plant.

The developed simulator generates the range of acceptable values of coefficients balancing the load of committing units with regard to ecological sources of energy.

The defined RER – renewable energy ratio - allows simple realization of generation policy in accordance with given contribution of renewable energy sources in the power plant set.

Economical effect of the technological structure of committing power plants on price of energy is calculated for the examined PS and the results are presented in the final paper.

REFERENCES

Baltierra A.E.; Moitre D.; Hernandez J.L.; Aromataris L. 1998. "Simulation of an Optimal Economic Strategy of a Wholesale Competitive Electric Energy Market". In *Proc. of 10th European Simulation Symposium*. Nottingham, England 1998. 255-259.

Kalinowski T., Malko J., Szalbierz Z., Wilczyński A. 1999. *Effectiveness of the international trade in the electric energy.* Kaprint Lublin. (in polish).

Malko J. 1997. Optimization of power generation structure on local market, Rynek Energii No. 5 (12), (in polish).

Sroczan E. 2006. "The simulation of the economic effect of power system structure including renewable sources of energy". In: *Modelling and Simulation ' 2006 . The 2006 Simulation and Modelling Conference.* Ed.: Nketsa A., Paludetto M., Bertelle C. Publ. EUROSIS-ETI. Ghent Belgium. 42-46.

Sroczan E. 2004. „Contemporary teleinformatic tools applied to energy management". *Rynek Energii*, Nr 1/50. 2-11 (in polish)

Sroczan E. 1996. "Application of Artificial Intelligence to Algorithms of Power Plant Identification with the Purpose of Load Dispatch". In *Proc.of 8th European Simulation Symposium ESS ' 96 "Genoa.* Society for Computer Simulation International. Genoa. 592-596.

EUGENIUSZ SROCZAN is employed as an assistant professor at the Poznan University of Technology (PUT) and professor of State Technical High School in Gniezno. Obtained from PUT a M.Sc. and Engineering Degree in area of Industry Automatic and a Ph.D. in area of Electric Power System Engineering. Author and co-author of papers on Power System Economic Operation, Energy Management Systems in Industry and Automation of Energetic Processes as well as Water and Waste-Water Treatment Plant. Author of the books on contemporary electrical installations of buildings. Since 1984 is the President of Branch of Association of Polish Electrical Engineers SEP at the PUT.

PROJECT MANAGEMENT SIMULATION

PROJECT MANAGEMENT PROCESSES MODELLING USING PETRI NETS

Sarka Kvetonova
Jitka Kreslikova
Faculty of Information Technology
Brno University of Technology
Bozetechova 2
612 66 Brno, Czech Republic
E-mail: {kvetona, kreslika}@fit.vutbr.cz

KEYWORDS

Project, Petri net, Project management, PMBOK, Process.

ABSTRACT

In the paper, a new approach of Petri Nets using to processes modelling of project management and subsequently to generalization of modelling abilities of Petri nets is presented. It describes how a technique of Petri Nets can be used to model and subsequent monitoring of software projects realization behaviour. The main purpose of this article is to simplify the process of project realization through creation and subsequent optimization of variant software project plans (especially for application in different domains of software engineering). These plans are created as Petri nets which allows us to optimize them by Petri Nets techniques and tools application. The possibility of project control processes verification especially with the knowledge of quality management field will contribute to the development of sophisticated Information Security Management System.

INTRODUCTION

Project management is applicable in all domains, in which the projects are realized. We perceive them as unique activities with hardly defined aim and determined terms, resources and scope. Each project is possible to be divided into a finite number of sub-processes, which are crucial to its complete successful realization. Of course, each project includes different types, categories and number of processes depending on which domain is project concerned, e.g. business systems, information systems, manufacture support systems etc. (Goodman 2006, Charvat 2003, Royce 2001)

In context of Petri nets we can use this technique to modelling of project processes, but without a necessity of separate approach. To date we have considered the Petri nets using to proposal, analysis and modelling of parallel and distribution systems both in technical and programming parties of computers. But we must still change and adapt the concrete Petri net to specific requirements of target users. This can be very prolonged and impractical. That is why we propose a new approach based on modelling of project processes. It is the first attempted generalisation of Petri Nets application into project management domain. There are the main ideas and thesis described and a simple example of Petri Nets using is presented.

In the following section the main foundations will be given that are crucial for expression and comprehension of a mutual context between terms of Project management and Petri Net.

TECHNICAL BACKROUND

Project management

Project management is a complex and extensive part of management. That is why we describe the basic terms, ideas and connections of it that are important for use the Petri nets in project management domain. A project management standard called PMBOK will be used as the main information source because of its detailed description and relating things of all processes of project management. The project processes are critical for each project, without reference to its orient, extent, duration etc. First, the important parts of project management will be described, after that the main processes of project management.

The basic terms of project management are as follow (Anbari 2005):

Project is a temporary effort undertaken to create a unique product or service, or result conforming to certain specifications and applicable standards.

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements. Project management is accomplished through the use of the processes such as: initiating, planning, executing, controlling, and closing.

PMBOK® Guide (A Guide to the Project Management Body of Knowledge) is a collection of processes and knowledge areas generally PM discipline and provides the fundamentals of PM that are applicable to a wide range of projects.

Process is a series of actions bringing about a result. It is a complex of mutually connected resources and activities, which changes inputs to outputs.

Project management involves the application of best practises to meet project objectives. The PMBOK Guide describes an integrative project management framework, which provides a basic structure for understanding project

management practices, composed of processes organized into process groups and knowledge areas.

Process groups are more formally known as project management process groups, knowledge areas are more formally known as project management knowledge areas, and each knowledge area is referenced as X management where X is the name of the knowledge area and it represents a subset of project management.

The basic concepts

Project management, projects, and the framework are integrative in that they require focusing on goals, balancing opposite demands, and leveraging feedback to ensure the goals are reached by given the opposite demands. The project triple constraint, which includes project scope, time, and cost parameters, provides a framework for evaluating opposite demands, integrating processes and their interactions. The project triple constraint is usually shown as a triangle where either each corner or edge represents one of the parameters that are managed. When we extend the iron project triangle to a polygon with combination of knowledge areas and process groups (described next), we get a picture introduced in Figure 1. Process groups are shown organized at the vertexes within the delimiting solid-outline square and include Initiating, Planning, Executing, Controlling, and Closing. The knowledge areas are shown organized within the solid-outline square around the project triple constraint triangle and include Integration, Scope, Time, Cost, Quality, Human Resource, Communications, Risk, and Procurement. Detailed description is given in the next part.

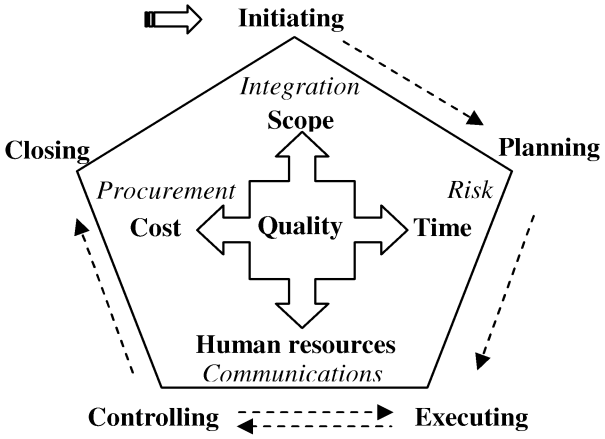


Figure 1: Process groups in a phase

The process groups are critical for our approach. In particular, we apply them to complex Petri Net model creation. It allows us to control whole project development process with respect to different levels of each project.

Categories of processes in project management

Project processes are executed by people and usually they fall into one of two major categories:

- *Project management processes* describe, organize, and complete the work of the project. For example PMBOK distinguishes thirty seven processes divided into nine knowledge areas. It is not necessary to implement all from those, it depends on requirement of management of concrete project or on extent of a project.
- *Product-oriented processes* specify and create the product of project. Product-oriented processes are typically defined by the project life cycle and vary by application area.

Project management processes and product-oriented processes overlap and interact throughout the project. For example, the scope of the project cannot be defined in the absence of some basic understanding of how to create the product.

Project management processes can be organized into five groups of one or more processes each (Project Management Institute 2000, Billows 2005):

- *Initiating processes* – authorizing the project or phase. This process group generally occurs at the beginning of a project or phase. The results from this group become input into the planning process group.
- *Planning processes* – defining and refining objectives and selecting the best of the alternative courses of action to attain the goals. This process group generally peeks early in a project or phase and steadily decreases. The results from this group become input into the Executing process group.
- *Executing processes* – coordinating people and other resources to carry out the plan. This process group generaly steadily increases and peeks later in a project or phase. The results from this process group become input into the Controlling process group.
- *Controlling processes* – ensuring that project objectives are met by monitoring and measuring progress regularly to identify variances from plan so that corrective action can be taken when necessary. This process group generaly steadily increases and peeks in the middle of a project or phase and steadily decreases. The results from this process group become input into the Planning, Executing, and Closing process group.
- *Closing processes* – formalizing acceptance of the project or phase and bringing it to an orderly end. This process group generaly occurs at the end of a project or phase. The results from this process group become input into the Initiating process group in the next project or phase.

The process groups are linked by the results they produce – the outcome of one often becomes an input to another.

A complex view on project management by PMBOK approach is captured *in Appendix 1*. There are mentioned all of project management domains in relations with other aspects of project management (business aims, strategies, mission etc.). We distinguish macro and micro level of the project management, too. Each of those levels consists of different sub-processes and individual activities. A concrete structure of those activities depends on a clearly defined business strategy, mission and from this derived business goals (strategic, operational, tactical).

The project management processes are common to most project in most application areas. Also the process interactions are typical of most projects in most application areas. That is why we have devoted to this problems and relations in detailed before.

PETRI NETS AND THEIR APPLICATIONS

A Petri net is essentially a bi-partite directed graph with annotations which represents a graphical and mathematical modelling tool targeted on modelling of distributed system structure. It consists of *places P* (represented as circles), *transitions T* (represented as bars), and *arcs* that connect them (but not transitions to transitions or places to places). *Input arcs* connect places with transitions, while *output arcs* start at a transition and end at a place. There are other types of arcs, e.g. *inhibitor arcs*. The dynamic behaviour of the system is usually represented by tokens flowing through the net. The current state of the modelled system (the *marking of the Petri net*) is given by the concrete number (and type if the tokens are distinguishable) of tokens in each place. The number of those tokens removed / added depends on the cardinality of each arc.

Transitions are active components. They model activities which can occur (the transition fires), thus changing the state of the system. Transitions are only allowed to fire if they are enabled, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places). When the transition fires, it removes tokens from its input places and adds some at all of its output places (Rozenberg 1991).

An example of Petri net is shown bellow (Figure 2):

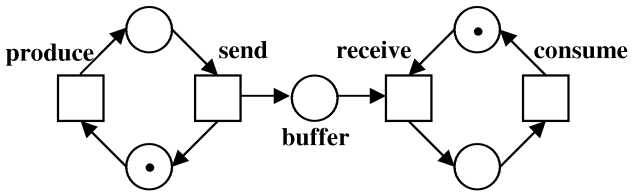


Figure 2: Petri net model of Producer and Consumer

As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behaviour of systems.

A formal definition of Petri nets is following (Penczek and Polrola 2006):

A Petri Net is a tetrad $N = (P, T, F, m^0)$, where
- $P = \{p_1,\ldots,p_n\}$ *is a finite set of* places,
- $T = \{t_1,\ldots,t_n\}$ *is a finite set of* transitions, *where* $P \cap T = \emptyset$ *and* $P \cup T \neq \emptyset$.
- F: $(P \times T) \cup (T \times P) \rightarrow IN$ *is the* flow function, *and*
- m^0: $P \rightarrow IN$ *is the* initial marking *of P.*

A variety of other formal definitions exist. Some of them do not include *arc weights* or *capacity restriction* (e.g. Murata 1989).

To study performance and dependability issues of systems it is necessary to include a timing concept into the model. There are several possibilities to do this for a Petri net; however, the most common way is to associate a *firing delay* with each transition. This delay specifies the time that the transition has to be *enabled*, before it can actually fire. If the delay is a random distribution function, the resulting net class is called *stochastic Petri net*. Different types of transitions can be distinguished depending on their associated delay, for instance *immediate transitions* (no delay), *exponential transitions* (delay is an exponential distribution), and *deterministic transitions* (delay is fixed).

There are many modifications and enlargements of Petri nets targeted on increase of its expression power, e.g. Timed Petri nets, Coloured Petri nets, Free Choice Petri nets, Hierarchical Petri nets etc.

OUR APPROACH

We propose a new approach to processes of project management modelling and their analysis. The project management needs a language that is intuitive and easy to use (as in the other domains of modelling).

At the same time, mere modelling of processes of project management is not sufficient. It is necessary to analyze the process model to look for possible flaws and further improvements. For analysis we need a more formal tool, and we gain that through Petri nets.

An example of project realization

The main information source is the PMBOK Standard. When we come out from it, we can define by this way all five process groups after that we can map them into Petri nets. After that individual process groups will be related together into one large Petri net. We will use one of the Petri nets method for this purpose targeted on modelling and analysing of complex systems or processes, in the concrete Hierarchical Petri net.

First, it is important to map all of activities and processes in accordance with PMBOK processes, activities and philosophy.

After that we organize it into one big Petri net. In this phase the project is divided into 2 parts:

1) primary individual activities/processes focused on project realization (relating to project management processes), and

2) secondary activities to support primary activities. It is necessary to document the whole process of the project realization. These types of documents include: input documentation (register of project, list of requests, contractual arrangement), reports of state, handing documentation for conclusion etc.

More complex processes including many sub processes are integrated into the model via hierarchization in Petri nets.

Created model (resultant Petri net) is shown in *Appendix 2*. Particular parts of Petri net are divided into process groups in respect to PMBOK standard (Initialization, Planning, Executing, Controlling and Closing processes). Activity "*Admin Closure*" is modeled as composed process including many particular processes. Concrete illustrations of dividing of these processes are given on the Figure 3. In this way it is possible to model highly problematic and complex projects or systems including many patterns, relationships and mutual associations (dependences), as well.

We can add various external triggers into Petri net. One of those triggers is shown in Appendix 2, too. An input into activity *Project Register* represents some important documentation necessary to realize the concrete task. The further possibility is a clock which can represent the time fixed on the actual situation in a project and finally an arrow depicts a necessity of a user input. We can put those triggers everyplace where the situation requires it.

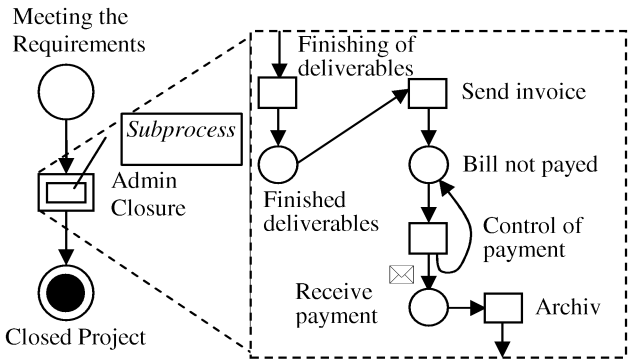


Figure 3: The hierarchical Petri net of „Admin Closure“ activity

CONCLUSION

There are a lot of publications and submissions on Petri nets theory. First of all Van Der Aalst is an author of many books focused on Petri nets application in specific modelling and analysing domains, for example workflow systems or systems engineering. There are a lot of enlargements of Petri nets like Petri nets with time, colour and hierarch to enhance their modelling power. The problem with this is that Petri nets are not an easy language for general modelling of processes without a binding on specific business domain. That is why we try to use this method for usage in project management domain, in other words we try to use Petri nets as an universal tool for modelling and analysing of project management processes, that we find almost everywhere and in the most of business domains.

Petri nets were chosen because of wide expressing power and thanks a lot of methods, techniques and tools that are available today. In the submission one of possible way of Petri nets application in project management domain is shown.

In the future, we suppose that the proposed model will be used for project planning support and will facilitate the whole process of the project realization by controlling all important sub-processes and their parties.

ACKNOWLEDGEMENTS

This research was supported by the Research Plan No. MSM 0021630528, Security-Oriented Research in Information Technology.

REFERENCES

Anbari, F. 2005. *Q & As for the PMBOK Guide*. Project Management Institute. ISBN 1930699395

Billows, D. 2005. *Project Management Best Practices.* Hampton Group. ISBN 0-9716820-3-8, 36-53.

Girault, C. 2003. *Petri Nets for Systems Engineering*. Springer. ISBN 3540412174

Goodman, A. 2006. *Process-based software project management.* AUERBACH. ISBN 0849373042.

Charvat, J. 2003. *Project management methodologics: selecting, implementing and supporting methodologies and processes for project.* New Jersey: John Wiley.

Murata, T. 1989. “Petri Nets: Properties, Analysis and Applications”. In *Proceedings of the IEEE*, Volume 77, Issue 4, 541-580.

Penczek, W. and Pólrola, A. 2006. *Advances in Verification of Time Petri Nets and Timed Automata. A Temporal Logic Approach.* Springer. ISBN 3-540-32869-6, 3-27.

Project Management Institute. 2000. *A Guide to the Project Management Body of Knowledge.* Project Management Institute. ISBN 1880410230

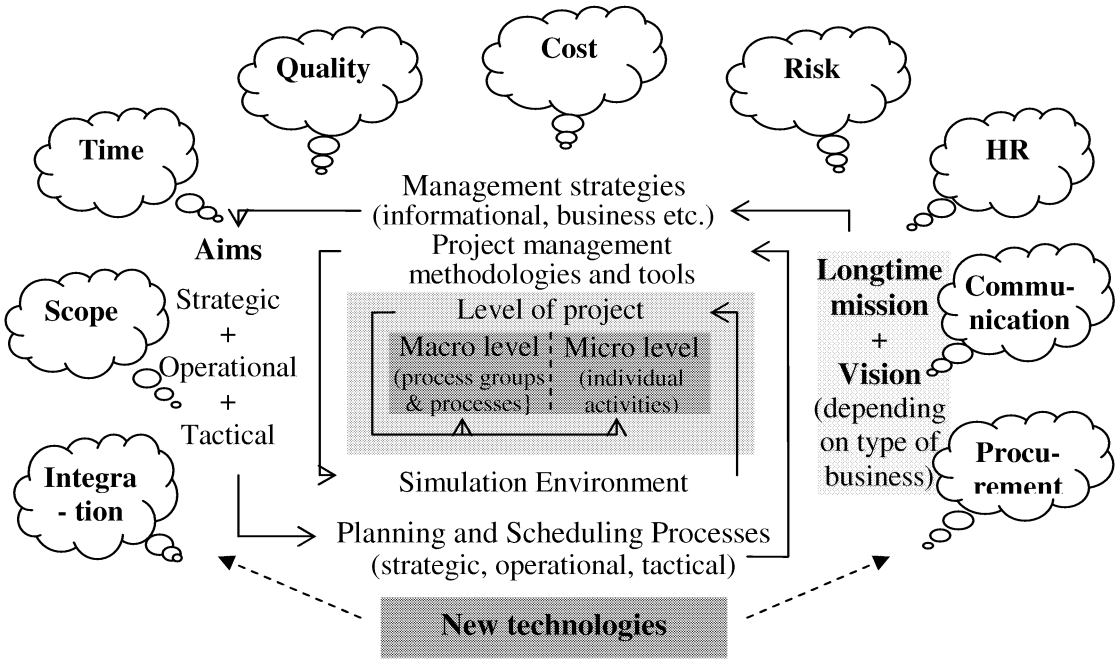
Royce, W. 2001. *Software project management: a unified framework.* PEARSON. ISBN 0201309580

Rozenberg, G. 1991. *Advances in Petri Nets.* Springer. ISBN 0387538631

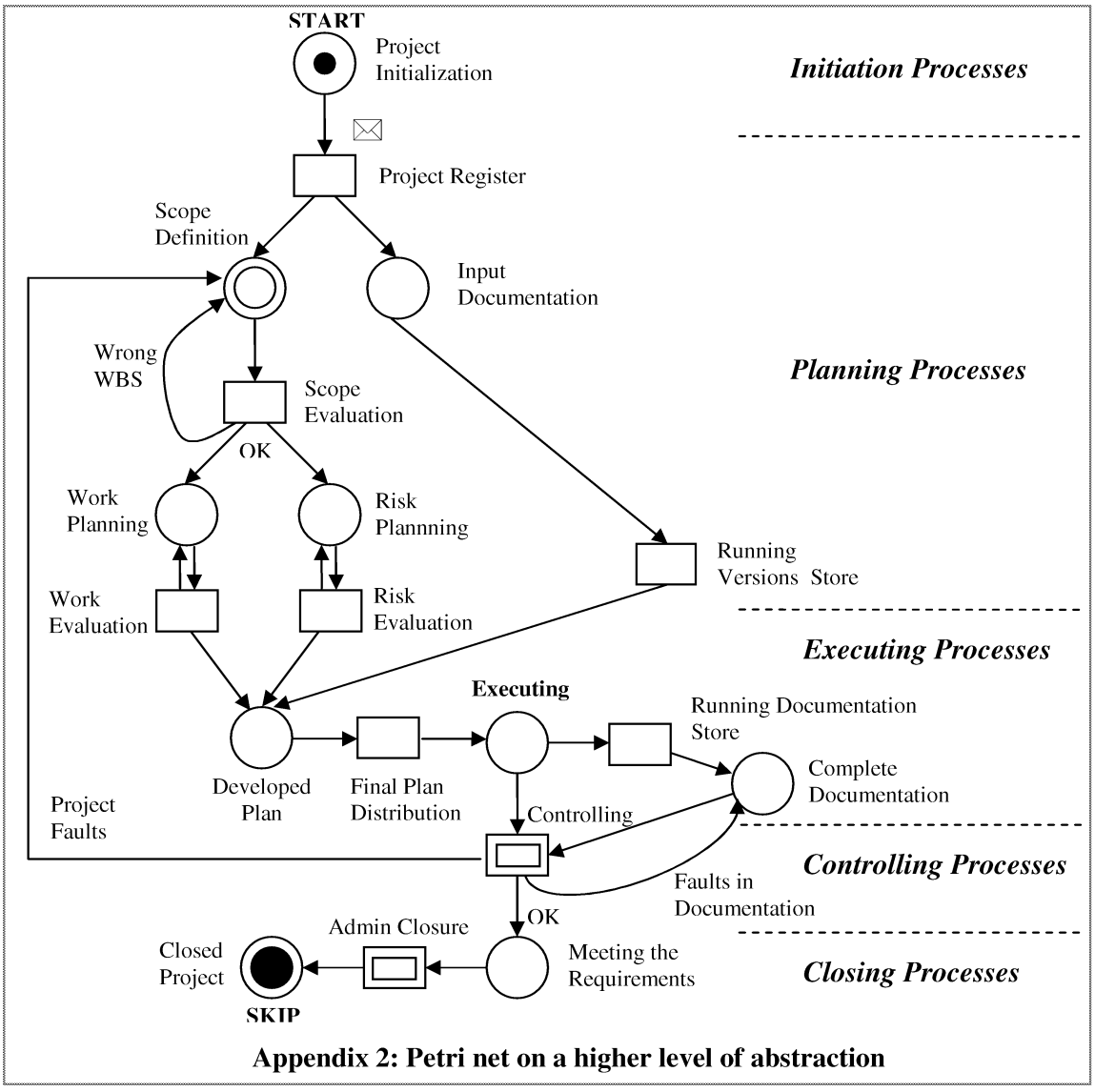
BIOGRAPHY

SARKA KVETONOVA was born in 1981, Brno, Czech Republic. She has studied Economics and Management at the Brno University of Technology, Faculty of Business and Management, Brno, Czech Republic. Presently, she is Ph.D. student at Brno University of Technology, Faculty of Information Technology, Brno, Czech Republic. Her main research interests are software engineering, processes and project management with focus on Petri Net techniques and their application for software projects management.

JITKA KRESLIKOVA was born in 1949, Czech Republic. She has studied Numerical Mathematics and Computer Science at the Masaryk University of Brno. She has achieved Ph.D. in 1993 in Applied Cybernetics, DCSE FEECS at Brno University of Technology, Czech Republic. Presently, she is assistant professor at Brno University of Technology, Faculty of Information Technology, Brno, Czech Republic. Her main research interests are Information Technology applied in Process Management, Project Management, Software Metrics and Quality areas.



Appendix 1: Complexity of project management by PMBOK approach



Appendix 2: Petri net on a higher level of abstraction

STRATEGIC MANPOWER PLANNING FOR THE BELGIAN DEFENCE: A MARKOVIAN STEADY-STATE MODEL

Filip Van Utterbeeck
Raf Van Loock

Department of Mathematics
Royal Military Academy
Renaissancelaan, 30
B-1000, Brussel
Belgium
e-mail: filip.van.utterbeeck@rma.ac.be

Dirk Van Oudheusden

Centre for Industrial Management
Katholieke Universiteit Leuven
Celestijnenlaan 300A
B-3001, Leuven
Belgium

KEYWORDS
Manpower Planning, Markov Chain, Mixed Career Concept, Defence

ABSTRACT

We develop a mathematical model for strategic manpower planning for the Belgian Defence. The model is based on stochastic Markov chain theory and can be used for long-term steady-state forecasting. It allows the estimation of the impact on future manpower requirements of demographic tendencies or changes in either the implemented human resource management policy or the legal framework governing military careers.

INTRODUCTION

On Jan 11th 2007 the Belgian parliament approved the law concerning the introduction of the so-called "Mixed Career Concept" for the Belgian Defence. This law encompasses a substantial modification of the legal statute governing the careers of the Belgian military personnel. The main aim of this new concept is the rejuvenation of the Belgian Armed Forces, in order to improve its operational capabilities and reduce the personnel cost. One of its key features is the introduction of so-called "mixed" careers: a significant percentage of the military personnel shall transfer to a civilian statute (in many cases government employed) after an initial military career of ten to fifteen years. This new concept heralds the abandonment of the principle of guaranteed life-long employment as a serviceman. The transition to the new statute is expected to take place around 2010 and involves several sets of transitional measures.

The model presented here is the first part of a set of decision making tools that are being developed to assist the human resource managers of the Belgian Defence. In future work, this steady-state model for long-term forecasting will be used in conjunction with a dynamic manpower planning model for short- and mid-term forecasting. The dynamic model will be based on stochastic discrete event simulation.

MATHEMATICAL MODEL

The approach presented here is based on the work of (Grinold and Marshall 1977) and (Bartholomew et al. 1991). (Wang 2005) presents an overview of OR techniques for military workforce planning and identifies four major techniques: Markov chain models, simulation models, optimisation models (including linear programming, goal programming and dynamic programming) and system dynamics models. Recent applications of manpower planning models can be found in (De Feyter 2007) and (Chattopadhyay and Gupta 2007).

We consider an organization with a manpower system consisting of N discrete *manpower classes* (n=1, 2, …, N), where n=0 refers to the external world (both the source for recruitment and the sink for people who leave the organization). We will observe the manpower system at certain *observation points* t (t=1, 2, …). We call the time interval between t-1 and t the *period* t. We define $s_i(t)$ as the *manpower stock* in class i at observation point t. The vector $s(t)=[s_1(t), s_2(t), \ldots, s_N(t)]$ gives the manpower stock of the complete organisation at observation point t. We further define the *flow* $f_{ij}(t)$ as the number of persons that belong to manpower class i at the start of period t and belong to manpower class j at the end of period t. We assume that this flow can be expressed as a fraction of the stock in class i at observation point t-1, and that this fraction is independent of both time and the manpower stock in class i:

$$f_{ij}(t) = q_{ji} s_i(t-1)\,, \qquad (1)$$

for i = 1, 2, …, N and j = 0, 1, …, N.

We can then write the manpower stock in function of the incoming flows:

$$s_j(t) = \sum_{i=0}^{N} f_{ij}(t) = f_{0j}(t) + \sum_{i=1}^{N} q_{ji} s_i(t-1)\,. \qquad (2)$$

We define $f_0(t)$ as the N-vector containing the recruitment flows $f_{0j}(t)$ and Q as the NxN-matrix containing the q_{ji}'s. When we write (2) in matrix form this gives:

$$s(t) = f_0(t) + Qs(t-1). \qquad (3)$$

The first term of (3) contains the recruitments while the second term contains the internal flows. This expression allows us to calculate the manpower stock at any observation point t if we know Q, the stock at an earlier observation point (typically $s(0)$) and the recruitment during the intervening periods.

We now consider that the size of our system remains constant and that all vacancies are filled at each observation point. We define the recruitment vector r as the Nx1-matrix $[r_1, r_2, \ldots, r_N]^T$, where r_j is the fraction of the total (external) recruitment that enters the system in class j. We also define the attrition vector a as the 1xN-matrix $[a_1, a_2, \ldots, a_N]$, where $a_j=q_{0j}$, the fraction of the manpower stock in class j that leaves the system during a given period. The number of vacancies created by persons leaving the system in period t now equals $\sum_{i=1}^{N} a_i s_i(t-1)$. As these are all filled, the external recruitment into each class can be written as follows:

$$f_{0j}(t) = r_j \sum_{i=1}^{N} a_i s_i(t-1). \qquad (4)$$

Inserting (4) into (2) gives:

$$s_j(t) = \sum_{i=1}^{N} (r_j a_i + q_{ji}) s_i(t-1), \qquad (5)$$

for j=1, 2, …,N. Finally, we define $P = r \cdot a + Q$ and we can write:

$$s(t) = Ps(t-1). \qquad (6)$$

Expression (6) has the same structure as the transition equations for a stochastic Markov chain. The matrix P is a stochastic matrix: all its elements are obviously non-negative and the column sums are all 1:

$$\sum_{j=1}^{N} p_{ji} = a_i \sum_{j=1}^{N} r_j + \sum_{j=1}^{N} q_{ji} = 1. \qquad (7)$$

(7) follows directly from $\sum_{j=1}^{N} r_j = 1$ and $a_i = q_{0i} = 1 - \sum_{j=1}^{N} q_{ji}$. The Perron-Frobenius theorem ensures that the largest eigenvalue associated with a stochastic matrix is always 1.

If our manpower system reaches steady-state, then the manpower stocks become time-independent and (6) becomes:

$$s = Ps. \qquad (8)$$

In other words: the steady-state distribution of the manpower stocks is given by the eigenvector of P associated with the eigenvalue 1. This eigenvector always exists, as P is a stochastic matrix. Multiplication of the eigenvector with the total manpower stock of the system yields the manpower in each class in steady-state.

IMPLEMENTATION

Introduction

We implemented the mathematical model described above in Matlab. We use Excel for the input and output, while the Excellink add-in (delivered with Matlab) assures the interface between Matlab and Excel. The utilisation of the tool can be subdivided into three phases, which are described below.

The length of the period t is set to 1 year. Each manpower class in the Belgian Defence model is characterized by the age, rank and seniority of its members. For instance all the 30-year old sergeants with 2 years of seniority in their rank share the same class. Some classes are further subdivided into subclasses: this is for instance the case for those classes where certain flows depend on the successful completion of an exam. The model is subdivided into 9 sub models representing coherent groups of manpower classes with few interactions. This was both done to diminish the run-time and to reflect the internal organisation of the manpower planning office.

Phase 1: defining the legal framework

The tool contains a separate Excel worksheet for every sub model which defines the legal framework. This sheet contains parameters such as for instance the minimum and maximum recruitment age, the retirement age, the minimum seniority in a given rank to be eligible for promotion, etc. The legal framework for the sub model for officers (not including the so-called specialist and expert careers) contains 63 parameters, for instance.

The worksheet contains a button which will create the parameter input sheet for phase 2. This involves the calculation of the maximum seniority attainable in every rank, as well as the minimum and maximum age within each combination rank-seniority, and the creation of a function which maps the triplet (rank, seniority, age) to (1, 2, …,N), with N being the number of manpower classes required to model the selected legal framework. The sub model for officers under the current legal framework for instance leads to N=8995. Obviously, although these classes are all theoretically possible, a lot of these classes will be rarely used.

Phase 2: setting the simulation parameters

Before the steady-state distribution can be calculated, the user has to define all the parameters governing the various flows in the system. We can distinguish two types of parameters:

- Policy parameters: these reflect options chosen by the human resource manager, most of these parameters govern the recruitment and promotion policy. These parameters represent the main degree of freedom for the users of the model.

- Environmental parameters: these reflect external factors which influence the manpower flows but which are largely outside the sphere of influence of the human resources manager, for example attrition due to personnel resignation or death and failure rates for formations and exams. These parameters are typically based on historical data, and must be estimated when such data is lacking.

The sub model for officers contains over 500 parameters, although a large part of these are optional (many parameters can be defined either for the entire system, by rank, or by combination of rank and seniority).

Upon completion of the parameter sheet, a simple button-click starts the calculation of Q, r and a in Matlab. These values are then used to calculate P and finally the eigenvector of P for eigenvalue 1. This eigenvector is then returned to excel in order to create the various output sheets.

One of the disadvantages of the modelling approach is that the model requires parameters governing flows to be defined as percentages of the manpower stock in a certain class (as shown by expression (1) above), while human resource managers prefer to think in absolute numbers. The parameter input sheet tries to accommodate this disadvantage by the inclusion of mirror-duplicates of every parameter table, which display the absolute values corresponding to every parameter after the execution of the model. Moreover, the model contains a tool which allows the iterative calculation of certain parameter values in order to attain desired absolute values.

Phase 3: output

The tool creates several output sheets in Excel, which provide summaries of the various stocks and flows, as well as derived information (such as for instance associated personnel costs), in both numerical and graphical formats. A separate post-hoc analysis sheet allows the user to impose limits on the manpower stocks in certain ranks or combination of ranks, an important feature as these limits are imposed by law. Figure 1 below shows a typical example of a graphical output for the officers sub model.

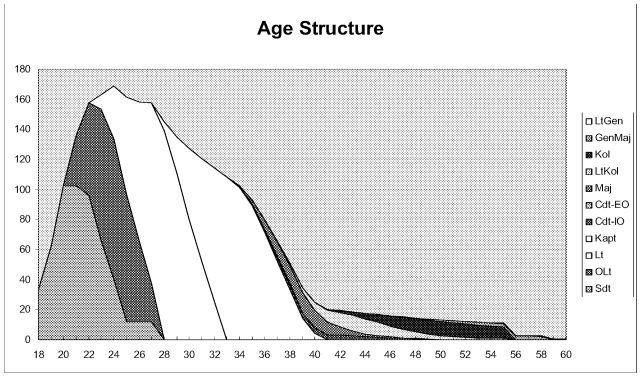


Figure 1: Steady State Manpower Stock Distribution by Age and Rank for Officers

CONCLUSIONS AND FUTURE WORK

We've created a tool for strategic manpower planning which allows the forecasting of the steady-state Belgian Defence manpower requirements under the new legal framework of the "mixed career concept". The user of the tool can modify a wide range of legal, environmental and policy parameters in order to provide answers to a wide range of "what if?" questions. The tool is based on a Markovian model approach which can easily be applied to other large hierarchical organisations.

In future work, we shall develop a dynamic manpower planning model which takes the actual manpower situation of the Belgian Defence as its input and which integrates the various transitional measures between the old statute and the statute of the mixed career concept. One of the aims of this model is the determination of the optimal policy in order to attain a state "close" to a predetermined steady-state distribution (calculated by the model described in this paper) within an imposed timeframe. This dynamic model shall be based on discrete event computer simulation coupled with heuristic optimization techniques.

REFERENCES

Bartholomew, D.J.; A.F. Forbes and S.I. McClean. 1991. *Statistical techniques for manpower planning*. John Wiley & Sons Ltd, Chichester.

Chattopadhyay, A.K. and A. Gupta. 2007. "A stochastic manpower planning model under varying class sizes." *Annals of Operations Research,* Vol. 155 No. 1, 41-49.

De Feyter, T. 2007. "Modeling mixed push and pull promotion flows in Manpower Planning." *Annals of Operations Research,* Vol. 155 No. 1, 25-39.

Grinold, R.C. and K.T. Marshall. 1977. *Manpower planning models*. North-Holland, New York.

Wang, J. 2005. "A review of operations research applications in workforce planning and potential modelling of military training." Technical report DSTO-TR-1688. Defence Science and Technology Organisation, Department of Defence, Australian Government.

BIOGRAPHY

FILIP VAN UTTERBEECK studied at the Royal Military Academy (RMA) in Brussels where he received the degree of polytechnical engineer in 1995. He worked as a technical officer and material manager in the Air Defence branch of the Belgian Air Force until 2002, when he returned to the RMA to become assistant lecturer at the Department of Mathematics. His main areas of interest are combinatorial optimization, statistics, discrete event simulation and their applications in human resource management and supply chain management. He currently pursues a PhD at the University of Leuven.

A BATCH SERVER WITH SERVICE TIMES DEPENDENT ON THE NUMBER OF SERVED CUSTOMERS

Dieter Claeys
Joris Walraevens*
Koenraad Laevens
Herwig Bruneel
SMACS Research Group
Department of Telecommunications and Information Processing
Ghent University
Sint-Pietersnieuwstraat 41
B 9000 Ghent, Belgium
E-mail:{dclaeys,jw,kl,hb}@telin.ugent.be
* The second author is a Postdoctoral Fellow with the Fund for Scientific Research, Flanders (F.W.O.-Vlaanderen), Belgium.

KEYWORDS
Queueing, performance analysis, transforms, stochastic, Markov chain.

ABSTRACT

Queueing models with a batch server operating under the minimum batch size policy have extensivily been studied in the past. In these models, the customers are served in batches of minimum l and maximum size c. However, to the best of the authors' knowledge, none of the models incorporate a dependency between the number of customers and the service time of the served batch, while in many practical applications (especially in telecommunication networks) this dependency is natural. In this paper, we contribute to filling this gap in the literature.
We derive the steady-state probability generating function of the number of customers in the system and we also obtain the probability generating function of the number of customers in a served batch in a single-server discrete-time queueing model with batch service.

INTRODUCTION

Servers that are capable of processing several customers simultaneously are regularly referred to as **batch servers**. The maximum number of customers that can be served simultaneously is called the server capacity. We remark that servers in traditional queueing models have a capacity of one. The service policy is an important aspect of batch servers. This service policy is a rule which determines when the server will start serving after it has become available. One possibility is to start service as soon as there is at least one customer in the queue. The disadvantage of this policy is that customers which arrive during a service period of a batch with few customers have to wait until the server becomes available again, since they cannot be added to the batch in service. Therefore, another extreme policy would be to wait until enough customers are present in the queue in order to serve at full capacity. However, this leads to an increasing delay of the already present customers. These policies are two extremes. A more general policy is the **minimum batch size** (MBS) policy introduced by (Neuts 1967): the batch server waits until the queue contents reaches or exceeds a certain threshold, before starting service. Choosing this threshold equal to one, yields the first extreme policy mentioned above, while a threshold equal to the server capacity corresponds to the other extreme.

Batch servers operating under the MBS service policy have been an important topic for research in the past (e.g. (Chaudhry and Templeton 1983), (Claeys et al. 2007), (Dümmler and Schömig 1999), (Goswami et al. 2006), (Gupta and Goswami 2002), (Powell and Humblet 1986), (Kim and Chaudhry 2006), (Samanta et al. 2007) and (Yi et al. 2007)). In all these studies, the authors assume that the service times are independent of the number of customers being served. However, in many practical situations the service times are dependent on the number of served customers. In this article, we want to fill this gap in the literature and study a discrete-time queueing model with a batch server and service times which have a general dependence on the number of served customers. This model is especially useful for telecommunication networks where some sort of aggregation of information units takes place at the sender's side. For example, in optical burst switched (OBS) networks (see e.g. (Qiao 1999)) information units are aggregated into an optical burst and then this burst is transmitted. It is logical that the more information units a burst contains, the longer the transmission lasts.

Below, we first describe our model in detail. Then, the steady-state probability generating function (PGF) of the system contents - the sum of the number of customers in the queue and the number of customers in the server - at the beginning of an arbitrary time slot is derived. After this, we determine the PGF of the number of customers

in a served batch. Next, we describe some numerical examples, which illustrate the influence of the introduced dependency. Finally, some conclusions are drawn.

MODEL DESCRIPTION

We consider the queueing model as depicted in Fig.1. This model has the following characteristics:

- The model is a discrete-time model, i.e. the time axis is divided into fixed-length periods, referred to as slots.
- During each slot, several customers can arrive. The number of customers that arrive during slot k is denoted by A_k. We assume that the sequence $\{A_k\}_{k\geq 1}$ consists of independent and identically distributed random variables, with common PGF $A(z)$. The number of customer arrivals during an arbitrary slot is denoted by A and has, in accordance with the previous assumptions, PGF $A(z)$.
- The queue is infinitely large. Therefore, all arriving customers can enter the queue and will eventually be served.
- Service is synchronized with respect to slot boundaries, i.e. the server always starts processing at the beginning of a slot and ends service at the end of a slot. This implies that a service period - the period between the start and end of the service of a batch - lasts an integer number of slots. We also assume that the service time - the length, in slots, of a service period - is dependent on the number of served customers. This assumption is a key feature of our queueing model. The service times are further independent of the previous service times and independent of the number of customer arrivals during the current slot and the previous slots. We denote the length of an arbitrary service period when the server serves j customers by S_j, with PGF $S_j(z)$.
- There is one batch server of capacity c and this server is operating under the MBS rule with threshold l, $1 \leq l \leq c$. So, when the server becomes available and finds less than l customers in the queue, the server waits to start processing until the beginning of the first slot whereby at least l customers are present. At the most, c customers can be served simultaneously. Furthermore, newly arriving customers cannot be added to the batch in service, though there might be free capacity.

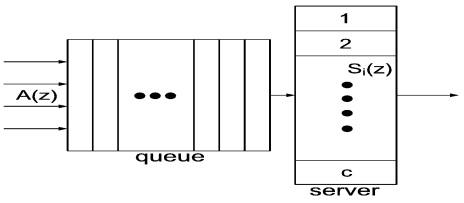


Figure 1: Model of the system, containing a queue and a batch server

The equilibrium condition of this queueing model requires that $A^{'}(1) < \frac{c}{S_c^{'}(1)}$, since if the system contents becomes large, batches of c customers will be served all the time. Note that we used the momentgenerating property of PGF's ($A^{'}(1) = \mathrm{E}[A]$ and $S_j^{'}(1) = \mathrm{E}[S_j]$) in our equilibrium condition and that we use primes to indicate derivatives.

GENERATING FUNCTION OF THE SYSTEM CONTENTS

Let us denote the system contents at the beginning of slot j by U_j and the system contents at the beginning of an arbitrary slot in the steady state by U. The sequence $\{U_j\}_{j\geq 1}$ does not form a Markov chain: besides the system contents, we have to keep track of the remaining service time if the server is processing. Furthermore, we also have to include the number of customers that are in service in our Markovian state description.

We denote the number of customers in the queue (server) at the beginning of slot k by Q_k (W_k). Note that $U_k = Q_k + W_k$. The remaining service time at the beginning of slot k is denoted by $H_k > 0$ if $W_k > 0$ and $H_k = 0$ if $W_k = 0$. Now, the sequence $\{Q_k, W_k, H_k\}_{k\geq 1}$ constitutes a Markov chain. If we observe the transition of the system when going from slot k to slot $k+1$, we can distinguish two cases, depending on the value of H_k:

- $H_k \leq 1$: this means that either the server is not processing during slot k ($H_k = 0$), because $U_k < l$, or the ongoing service is finished at the end of slot k ($H_k = 1$). The system contents at the beginning of slot $k+1$ is equal to $Q_k + A_k$. Now, we distinguish 3 subcases depending on the value of $E_k \equiv Q_k + A_k$:
  - $E_k < l$: there are not enough customers in the system to start service in slot $k+1$. This implies that $Q_{k+1} = E_k$, $W_{k+1} = 0$ and $H_{k+1} = 0$.
  - $l \leq E_k < c$: in this case the server starts processing all the customers in the system at the start of slot $k+1$. So, $Q_{k+1} = 0$, $W_{k+1} = E_k$ and $H_{k+1} = S_{k+1,E_k}$, with $S_{n,m}$ the length of the service period starting at slot n and whereby m customers are served. With the previous assumptions $S_{n,m}$ has the same distribution as S_m.
  - $E_k \geq c$: the server starts serving the c first customers and the other customers remain in the queue. We thus have: $Q_{k+1} = E_k - c$, $W_{k+1} = c$ and $H_{k+1} = S_{k+1,c}$.
- $H_k > 1$: in this case the server is processing during slot k and the service period does not end at the end of slot k. In this case, $Q_{k+1} = E_k$, $W_{k+1} = W_k$ and $H_{k+1} = H_k - 1$.

We denote the joint PGF of Q_k, W_k and H_k by $V_k(z, x, y)$. Note that $U(z)$ - the steady-state PGF of the system contents at the beginning of an arbitrary slot - is equal to $V(z, z, 1)$, since the system contents is equal to the sum of the queue contents and the number of customers in the server and since we "sum out" H_k if we substitute y by 1. Translating the previous system equations to PGF's and

introducing the notation $\mathrm{E}\left[z^X\{Y\}\right] \equiv \mathrm{E}\left[z^X|Y\right]\Pr[Y]$, we find:

$$\begin{aligned} V_{k+1}(z,x,y) &= \mathrm{E}\left[z^{Q_{k+1}}x^{W_{k+1}}y^{H_{k+1}}\right] \\ &= \mathrm{E}\left[z^{E_k}\{H_k \le 1, E_k < l\}\right] \\ &\quad +\mathrm{E}\left[x^{E_k}y^{S_{E_k}}\{H_k \le 1, l \le E_k < c\}\right] \\ &\quad +\frac{x^c S_c(y)}{z^c}\mathrm{E}\left[z^{E_k}\{H_k \le 1, E_k \ge c\}\right] \\ &\quad +\frac{1}{y}\mathrm{E}\left[z^{E_k}x^{W_k}y^{H_k}\{H_k > 1\}\right] \quad . \end{aligned} \tag{1}$$

Now, we calculate the terms of (1) separately. First, we observe that $\lim_{k\to\infty}\mathrm{E}\left[z^{E_k}\{H_k \le 1, E_k < l\}\right]$ is a polynomial of degree $l-1$. Thus, we can write

$$\lim_{k\to\infty}\mathrm{E}\left[z^{E_k}\{H_k \le 1, E_k < l\}\right] = \sum_{n=0}^{l-1} e(n)z^n \quad , \tag{2}$$

with $e(n) \equiv \lim_{k\to\infty}\Pr[E_k = n, H_k \le 1]$. We further have that

$$\begin{aligned} &\lim_{k\to\infty}\mathrm{E}\left[x^{E_k}y^{S_{E_k}}\{H_k \le 1, l \le E_k < c\}\right] \\ &= \sum_{n=l}^{c-1} e(n)x^n S_n(y) \quad . \end{aligned} \tag{3}$$

Furthermore, we also obtain that

$$\begin{aligned} &\lim_{k\to\infty}\mathrm{E}\left[z^{E_k}\{H_k \le 1, E_k \ge c\}\right] \\ &= E(z) - \sum_{n=0}^{l-1} e(n)z^n - \sum_{n=l}^{c-1} e(n)z^n \quad , \end{aligned} \tag{4}$$

with $E(z) = \sum_{n=0}^{\infty} e(n)z^n$. Finally, if we introduce the notations $R_k(z,x) \equiv \mathrm{E}\left[z^{Q_k}x^{W_k}\{H_k = 1\}\right]$ and $q_{k,0}(n) \equiv \Pr[Q_k = n, H_k = 0]$, the last term of (1) becomes:

$$\frac{A(z)}{y}\left[V_k(z,x,y) - \sum_{n=0}^{l-1} q_{k,0}(n)z^n - yR_k(z,x)\right] \quad , \tag{5}$$

since A_k is independent of W_k and H_k and $q_{k,0}(n) = 0$ if $n \ge l$. The combination of (1) and (2) - (5) yields (in the steady state):

$$\begin{aligned} V(z,x,y) &= \lim_{k\to\infty} V_k(z,x,y) \\ &= \left[1 - \frac{x^c}{z^c}S_c(y)\right]\sum_{n=0}^{l-1} e(n)z^n \\ &\quad +\sum_{n=l}^{c-1} e(n)\left[x^n S_n(y) - \frac{x^c}{z^c}S_c(y)z^n\right] \\ &\quad +\frac{x^c S_c(y)}{z^c}E(z) - A(z)R(z,x) \\ &\quad +\frac{A(z)}{y}\left[V(z,x,y) - \sum_{n=0}^{l-1} q_0(n)z^n\right] \quad , \end{aligned} \tag{6}$$

with $R(z,x) \equiv \lim_{k\to\infty} R_k(z,x)$ and $q_0(n) \equiv \lim_{k\to\infty} q_{k,0}(n)$. We remark that

$$E(z) = A(z)\left[\sum_{n=0}^{l-1} q_0(n)z^n + R(z,1)\right] \quad . \tag{7}$$

Multiplying formula (6) by y and substituting y by 0, we find:

$$\sum_{n=0}^{l-1} q_0(n)z^n = \sum_{n=0}^{l-1} e(n)z^n \quad , \tag{8}$$

since $V(z,x,0) = \sum_{n=0}^{l-1} q_0(n)z^n$. Combining (6) with (7) and (8) and rearranging the terms, we can write:

$$\begin{aligned} &\left[1 - \frac{A(z)}{y}\right]V(z,x,y) \\ &= \left[1 - \frac{A(z)}{y}\right]\sum_{n=0}^{l-1} q_0(n)z^n \\ &\quad +\frac{x^c}{z^c}S_c(y)(A(z)-1)\sum_{n=0}^{l-1} q_0(n)z^n \\ &\quad +\frac{x^c}{z^c}S_c(y)A(z)R(z,1) \\ &\quad -A(z)R(z,x) \\ &\quad +\sum_{n=l}^{c-1} e(n)\left[x^n S_n(y) - \frac{x^c}{z^c}S_c(y)z^n\right] \quad . \end{aligned} \tag{9}$$

As mentioned above, the steady-state PGF of the system contents at the beginning of an arbitrary slot ($U(z)$) can be obtained by replacing x by z and y by 1 in (9). We obtain:

$$[1 - A(z)]\,U(z) = A(z)R(z,1) - A(z)R(z,z) \quad . \tag{10}$$

It is obvious that we still have to calculate $A(z)R(z,1)$ and $A(z)R(z,z)$. The substitution of y by $A(z)$ and x by 1 in formula (9) provides $A(z)R(z,1)$ as a function of $q_0(n)$ $(0 \le n \le l-1)$ and $e(n)$ $(l \le n \le c-1)$:

$$\begin{aligned} A(z)R(z,1) &= \left[S_c(A(z))(A(z)-1)\sum_{n=0}^{l-1} q_0(n)z^n\right. \\ &\quad \left.+\sum_{n=l}^{c-1} e(n)\left[S_n(A(z))z^c - S_c(A(z))z^n\right]\right] \\ &\quad \times\frac{1}{z^c - S_c(A(z))} \quad . \end{aligned} \tag{11}$$

Analogously, we find the following expression for $A(z)R(z,z)$ by substituting y by $A(z)$ and x by z in (9) and by using expression (11) to replace $R(z,1)$:

$$\begin{aligned} A(z)R(z,z) = &\quad \frac{1}{z^c - S_c(A(z))} \\ &\left[z^c S_c(A(z))(A(z)-1)\sum_{n=0}^{l-1} q_0(n)z^n\right. \\ &\left.+\sum_{n=l}^{c-1} e(n)f_n(z)\right] \quad , \end{aligned} \tag{12}$$

with

$$\begin{aligned} f_n(z) &= \left[S_n(A(z)) - S_c(A(z))\right]\left[z^c - S_c(A(z))\right]z^n \\ &\quad +S_c(A(z))\left[S_n(A(z))z^c - S_c(A(z))z^n\right] \quad . \end{aligned}$$

We finally obtain an expression for $U(z)$ by combining (10), (11) and (12):

$$\begin{aligned} U(z) = &\quad \frac{1}{z^c - S_c(A(z))} \times \\ &\left[S_c(A(z))(z^c - 1)\sum_{n=0}^{l-1} q_0(n)z^n\right. \\ &\left.+\frac{1}{1 - A(z)}\sum_{n=l}^{c-1} e(n)g_n(z)\right] \quad , \end{aligned} \tag{13}$$

with

$$\begin{aligned} g_n(z) &= S_n(A(z))z^c\left[1 - S_c(A(z)) - z^n\right] \\ &\quad -S_c(A(z))z^n\left[1 - S_n(A(z)) - z^c\right] \quad . \end{aligned} \tag{14}$$

The expression for $U(z)$ still contains c unknown constants ($q_0(n)$, $0 \le n \le l-1$ and $e(n)$, $l \le n \le c-1$). These can be determined by solving a set of c linear equations: by means of Rouché's theorem, we can prove that the denominator of the right-hand side of (13) has

c zeroes ($z_0 = 1, z_1, \ldots, z_{c-1}$) inside and on the complex unit disk $\{|z| \leq 1\}$. Due to the analytic nature of PGF's in that area, these zeroes also have to be zeroes of the numerator. Our set of linear equations then consists of the normalization condition $U(1) = 1$ and the $c-1$ equations $N(z_i) = 0, 1 \leq i \leq c-1$, with $N(z) = S_c(A(z))(z^c - 1)\sum_{n=0}^{l-1} q_0(n)z^n + \frac{1}{1-A(z)}\sum_{n=l}^{c-1} e(n)g_n(z)$.

Remark 1 *The formula for $U(z)$ for the batch-server model with service times independent of the number of served customers can be obtained from (13) and (14) by substituting all $S_i(z)$ by (a common) PGF $S(z)$:*

$$\begin{aligned} U(z) &= \frac{S(A(z))}{z^c - S(A(z))} \times \\ &\quad \left[(z^c - 1)\sum_{n=0}^{l-1} q_0(n)z^n \right. \\ &\quad \left. + S^*(A(z))\sum_{n=l}^{c-1} v(n)(z^c - z^n) \right] , \end{aligned}$$

where we have introduced $v(n) = S^{'}(1)e(n)$ and where $S^*(z) = \frac{S(z)-1}{S^{'}(1)(z-1)}$ is the PGF of the position of an arbitrary slot in the service period; the first slot of a service period is assumed to have position 0. We have indeed found this expression in (Claeys et al. 2007).

Remark 2 *We can also calculate other PGF's from expression (9), such as the PGF of the queue contents. In the next section we derive, starting from (9), the PGF of the number of customers in a served batch.*

PGF OF THE BATCH SIZE

In this section, we calculate the PGF of the number of customers being served during a random service period. Let us denote the probability that n customers ($l \leq n \leq c$) are served in a random service period by $b(n)$, with PGF $B(x)$. If we substitute z by 1 in (9), multiply both sides by y and substitute y by 1, we have:

$$R(1,x) = x^c R(1,1) + \sum_{n=l}^{c-1} e(n)\left[x^n - x^c\right] .$$

Since the probability of serving a batch of n customers is equal to the probability that the server contains n customers during a randomly chosen last slot of a service period, we can see that $R(1,x) = B(x)R(1,1)$ and we thus obtain:

$$B(x) = x^c + \sum_{n=l}^{c-1} \frac{e(n)}{R(1,1)}\left[x^n - x^c\right] . \quad (15)$$

From (15) we can directly "read" the probabilities $b(n)$:

$$b(n) = \begin{cases} \frac{e(n)}{R(1,1)} & \text{if } l \leq n \leq c-1 \\ 1 - \frac{\sum_{n=l}^{c-1} e(n)}{R(1,1)} & \text{if } n = c \\ 0 & \text{else} \end{cases}$$

We calculate $R(1,1)$ by substituting z by 1 in (12). After using de l'Hospital's rule, we find

$$\begin{aligned} R(1,1) &= \frac{1}{c - S_c^{'}(1)A^{'}(1)} \times \\ &\quad \left[A^{'}(1)\sum_{n=0}^{l-1} q_0(n) + \sum_{n=l}^{c-1} e(n) \times \right. \\ &\quad \left. \left\{ A^{'}(1)\left(S_n^{'}(1) - S_c^{'}(1)\right) + (c-n) \right\} \right] . \end{aligned}$$

Remark 3 *We cannot easily read the probabilities $u(n)$ of having n customers in the system at the beginning of an arbitrary slot. However, the $u(n)$ can be obtained by several methods, such as e.g. the probability generating property of PGF's $\left(u(n) = \frac{1}{n!}\frac{d^n U(z)}{dz^n}\right)$, tail approximations (see e.g. (Bruneel et al. 1994)), ... We do not discuss this in more detail. In the numerical examples in the next section, we are mainly interested in the mean value of U and we obtain this by means of the moment generating property of PGF's ($\mathrm{E}[U] = U^{'}(1)$). From $\mathrm{E}[U]$ we can calculate the mean delay of customers - defined as the total number of slots between the end of the customer's arrival slot and the end of the customer's departure slot - by means of the discretised version of Little's theorem (see e.g. (Fiems and Bruneel 2002)).*

NUMERICAL EXAMPLES

In this section, we study the influence of the dependency of the service times on the numbers of served customers. We do this by comparing the results for two models. In the first model, the service times are not dependent on the number of served customers. We there assume that $S_i(z) = \frac{0.1z}{1-0.9z} \equiv S(z), l \leq i \leq c$, i.e. the service times are geometrically distributed, with a mean value of 10 slots. In the second model, we assume $S_i(z) = \frac{(1-\alpha_i)z}{1-\alpha_i z}$, with $S_i^{'}(1) = \frac{1}{1-\alpha_i} = iK + 1$, where K is a scale factor so that $\frac{\sum_{n=l}^{c} S_n^{'}(1)}{c-l+1} = 10$. So, in this case the service times are larger if the server serves more customers. The conditional service times are still geometrically distributed and the mean of the conditional mean values is equal to 10 slots.

Figure 2 shows the mean system contents in terms of $A^{'}(1)$. The left (right) pane corresponds to the first (second) model. Analogously, Fig. 3 shows the mean delay for these models. We assumed a Poisson distribution for the number of customer arrivals during a slot. The server has a capacity of $c = 10$. We plotted the curves for 3 different values of the MBS ($l = 1$, $l = 5$ and $l = 10$). In general, we see a large difference between the results for both models. For low $A^{'}(1)$ and $l < c$, we observe that the second model performs better (i.e. a smaller mean system contents and thus also a smaller mean delay). This is due to the fact that in this case the minimal number of l customers are served most of the time and since $S_l^{'}(1) < S^{'}(1)$. We also observe that the mean system contents and delay increase faster for the second model, because larger $A^{'}(1)$ means that a served batch contains more customers and these service times are longer so the server becomes less available. This implies that the system becomes unstable for a lower $A^{'}(1)$. Furthermore, we see that the stability bound appears for a lower $A^{'}(1)$ if l is lower. This is a consequence of the fact that the $S_c^{'}(1)$ is larger if l is smaller and since the stability condition solely depends on $A^{'}(1)$ and $S_c^{'}(1)$. Finally, the curve for $l = 10$ remains the same. Indeed, in this case, the server only serves full batches and $S_c^{'}(1) = S^{'}(1)$.

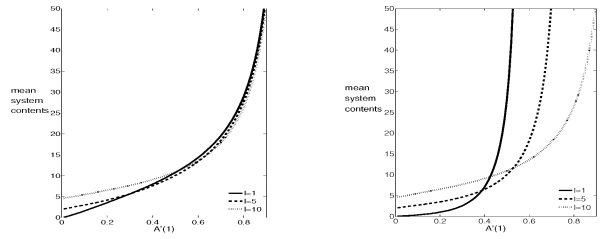


Figure 2: Comparison of mean system contents; left pane: independent model; right pane: dependent model; $c = 10$; $S^{'}(1) = 10$; Poisson arrivals

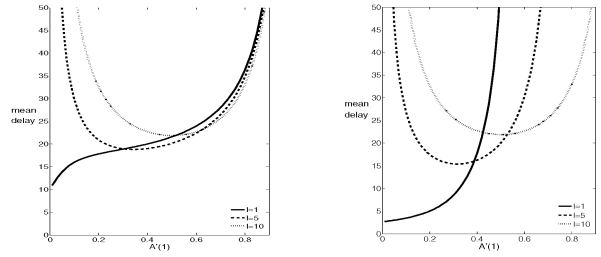


Figure 3: Comparison of mean delay; left pane: independent model; right pane: dependent model; $c = 10$; $S^{'}(1) = 10$; Poisson arrivals

Now, we take a closer look at the probabilities $b(n)$ of serving n customers in a random service period. Table 1 compares the $b(n)$ of both models for several values of $A^{'}(1)$. We choose $c = 10$, $l = 5$, $S^{'}(1) = 10$ and Poisson number of arrivals per slot. Note that for these choices, we cannot get accurate results for the second model if $A^{'}(1) > 0.65$. This is because of the stability bound that appears for $A^{'}(1)$ around 0.65 (see right pane of figures 2 & 3). For both models we observe, as expected, that, in case of low $A^{'}(1)$, $b(l)$ is largest and, in case of large $A^{'}(1)$, $b(c)$ is largest. If we compare the second model to the first model, we observe that, for low $A^{'}(1)$, $b(l)$ is larger for the second model. This is because a low $A^{'}(1)$ means a higher probability of serving l customers and for the second model this service time is shorter which in turn implies that at the end of the service time less customers have arrived. So, the probability of serving l customers again at the beginning of the next service period is larger. Analogously, we find that $b(c)$ is larger for the second model than for the first model if $A^{'}(1)$ is large.

Table 1: $b(n)$ for both models with c=10, $l = 5$ and $S^{'}(1) = 10$ for several values of $A^{'}(1)$; left (right) part of each column: first model (second model)

n	$A^{'}(1) = 0.1$		$A^{'}(1) = 0.3$		$A^{'}(1) = 0.65$	
5	0.939	0.948	0.708	0.775	0.306	0.277
6	0.053	0.049	0.140	0.137	0.126	0.108
7	0.005	0.003	0.045	0.033	0.063	0.049
8	0.002	0.000	0.027	0.017	0.046	0.033
9	0.001	0.000	0.020	0.011	0.040	0.028
10	0.001	0.000	0.061	0.027	0.420	0.506

CONCLUSIONS

In this paper, we have extended the traditional model of a batch server operating under the MBS service policy with a dependency of the service time on the number of customers in the served batch. For this model, we have derived the steady-state PGF of the system contents at the beginning of an arbitrary slot. We also have obtained the PGF of the number of customers in a served batch. Furthermore, we have compared this dependent model with the independent model. We can conclude that the behavior in terms of the mean system contents, delay and number of served customers can be highly dependent on this dependency and ignoring it may lead to a high inaccuracy of the results.

In the future, we plan to consider other examples of dependencies between the number of served customers and the service times. Furthermore, we plan to incorporate some kind of correlation in the arrival process of our queueing model.

REFERENCES

Bruneel, H.; Steyaert, B.; Desmet, E. and Petit, G.H. 1994. "Analytic derivation of tail probabilities for queue lengths and waiting times in ATM multiserver queues." *European Journal of Operational Research*, No.4, 563–572.

Chaudhry, M.L. and Templeton, J.G.C. 1983. *A First Course in Bulk Queues*. Wiley, New York.

Claeys, D.; Walraevens, J.; Laevens, K. and Bruneel, H. 2007. "A Discrete-time Queueing Model with a Batch Server Operating under the Minimum Batch Size Rule". *Proceedings of the 7th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking* (St. Petersburg, Sept.10–14). Lecture Notes in Computer Science 4712, 248–259.

Dümmler, M.A. and Schömig, A.K. 1999. "Using Discrete-time Analysis in the Performance Evaluation of Manufacturing Systems". *Proceedings of the International Conference on Semiconductor Manufacturing Operational Modeling and Simulation* (San Francisco, California, Jan.18–20).

Fiems, D. and Bruneel, H. 2002. "A note on the discretization of Little's result". *Operations Research Letters*, No.30, 17–18.

Goswami, V.; Mohanty, J.R. and Samanta, S.K. 2006. "Discrete-time bulk-service queues with accessible and non-accessible batches". *Applied Mathematics and Computation*, No.182, 898–906.

Gupta, U.C. and Goswami, V. 2002. "Performance analysis of finite buffer discrete-time queue with bulk service". *Computers & Operations Research*, No.29, 1331–1341.

Powell, W.B. and Humblet, P. 1986. "The bulk service queue with a general control strategy: theoretical analysis and a new computational procedure". *Operations Research*, No.34(2), 267–275.

Kim, N.K. and Chaudhry, M.L. 2006. "Equivalences of Batch-Service Queues and Multi-Server Queues and Their Complete Simple Solutions in Terms of Roots". *Stochastic Analysis and Applications*, No.24, 753–766.

Neuts, M.F. 1967. "A general class of bulk queues with Poisson input". *Ann. Math. Stat.*, No. 38, 759–770.

Qiao, C.M and Yoo, M.S. 1999. Optical burst switching (OBS) - a new paradigm for an optical Internet. *Journal of high speed networks*, No.8(1), 69–84.

Samanta, S.K.; Chaudhry, M.L. and Gupta, U.C. 2007. "Discrete-time $Geo^X|G^{(a,b)}|1|N$ queues with single and multiple vacations". *Mathematical and Computer Modelling*, No.45, 93–108.

Yi, X.W.; Kim, N.K.; Yoon, B.K and Chae, K.C. 2007. "Analysis of the queue-length distribution for the discrete-time batch-service $Geo^X|G^{a,Y}|1|K$ queue". *European Journal of Operational Research*, No.181, 787–792.

BIOGRAPHY

Dieter Claeys was born in Kortrijk, Belgium. He received the degree of Licentiate in Computer Sciences in 2006, from Ghent University, Belgium. After his studies, he has been working as a Ph.D. student for the SMACS research group of the Department for Telecommunications and Information Processing at the Ghent University. The SMACS research group is specialised in the statistic modeling and analysis of communication systems. The main topic of Dieter's research is the analysis of discrete-time queueing models with batch service.

Joris Walraevens received the M.S. degree in Electrical Engineering and the Ph.D. degree in Engineering in 1997 and 2004 respectively, all from Ghent University, Belgium. In September 1997, he joined the SMACS Research Group, Department for Telecommunications and Information Processing, at the same university. Since October 2006, he is a postdoctoral fellow with the Fund for Scientific Research, Flanders (F.W.O.-Vlaanderen), Belgium. His main research interests include discrete-time queueing models and performance analysis of communication networks.

Koenraad Laevens obtained a MSc. in Electrical Engineering and a Ph.D. in Applied Sciences from Ghent University (Belgium), in 1991 and 1999 respectively. During the years, he held several positions as researcher at Ghent University, Microsoft Research Cambridge, and Alcatel-Lucent Bell (formerly Alcatel Bell). His research focusses on the broad area of performance evalution, emphasizing applications in the telecom sector. His current interests include congestion control and video delivery mechanisms in packet-switched networks.

Herwig Bruneel was born in Zottegem, Belgium, in 1954. He received the MS degree in Electrical Engineering, the degree of Licentiate in Computer Science, and the Ph.D. degree in Computer Science in 1978, 1979 and 1984, respectively, all from Ghent University, Belgium. He is full time Professor in the Faculty of Applied Sciences and Head of the Department of Telecommunications and Information Processing at the same university. He also leads the SMACS Research Group within his department. His main personal research interests include stochastic modeling and analysis of communication systems, discrete-time queueing theory, and the study of ARQ protocols. He has published more than 200 papers on these subjects and is co-author of the book by H. Bruneel and B.G. Kim, "Discrete-Time Models for Communication Systems Including ATM" (Kluwer Academic Publishers, Boston, 1993). From October 2001 to September 2003, he served as the Academic Director for Research Affairs at Ghent University.

TWO-BAND k·p MODEL FOR THE CONDUCTION BAND IN SILICON

Viktor Sverdlov, Gerhard Karlowatz, Hans Kosina, and Siegfried Selberherr
Institute for Microelectronics
Vienna University of Technology
Gußhausstraße 27–29/E360, A-1040 Wien, Austria
e-mail: sverdlov@iue.tuwien.ac.at

KEYWORDS
Silicon band structure, **k·p** theory, mobility, Monte Carlo

ABSTRACT

The modification of the silicon conduction band under uniaxial [110] stress is considered. Special attention is paid to the stress dependence of the non-parabolicity parameter. An analytical expression of the dependence of the non-parabolicity parameter on shear stress is obtained. At 3 GPa stress the non-parabolicity is shown to increase by a factor of 1.7. The stress dependence of the non-parabolicity parameter is verified by comparing the corresponding analytical density-of-states to the numerical density-of-states obtained from the empirical pseudopotential method and good agreement is found. The increase in the non-parabolicity parameter increases the after-scattering density-of-states and the scattering rates, which results in an almost 25% suppression of the mobility enhancement due to the corresponding effective mass decrease in a 3 nm Si body thin FET at 3 GPa [110] stress.

INTRODUCTION

Stress-induced mobility engineering has become a key technique to increase the performance of modern CMOS devices. The reason for the mobility enhancement lies in the band structure modification caused by stress. The conduction band in Si is commonly approximated by three pairs of equivalent valleys with their minima located close to the X-points of the Brillouin zone. Close to the minima the electron dispersion is well described by the effective mass approximation. At higher electron energies the non-parabolicity parameter (typically $\alpha = 0.5$ eV^{-1}) has to be introduced to describe deviations from the density of states of the purely parabolic dispersion.

The semiconductor industry is exploiting stress techniques compatible with existing CMOS process technology. Stress is generated by local stressors and/or additional cap layers. The technologically relevant stress along the [110] direction was systematically investigated experimentally only recently [1]. Inherent to [110] uniaxial stress is a shear distortion of the Si crystal lattice, which induces pronounced modifications in the conduction band. Shear strain produces nonlinear shifts between the valleys [2, 3], and the degeneracy between the six equivalent valleys is lifted. Shear strain also modifies substantially both the longitudinal [2, 3] and transversal [1, 2, 3, 4] effective masses. The transversal mass determines the mobility in a FET with ultra-thin Si body. In such FETs the electron mobility enhancement induced by [110] tensile stress is therefore solely due to a decrease of the conductivity mass in the stress direction [1, 2, 3]. However, a possible stress dependence of the non-parabolicity parameter has not yet been considered.

TWO-BAND k·p THEORY AND NON-PARABOLICITY

We consider the valley along the [001] direction only, other valleys are analyzed in a similar fashion. The closest band to the first conduction band Δ_1 ($i = 1$) is the second conduction band $\Delta_{2'}$ ($i = 2$). The two bands become degenerate exactly at the X points. Since the minimum of the conduction band is only $k_0 = 0.15\frac{2\pi}{a}$ away from the X point, the dispersion around the minimum is well described by degenerate perturbation theory, which includes only these two bands. Diagonal elements of the Hamiltonian H_{ii}at the X point, where $i = 1, 2$ are:

$$H^0_{ii}(k) = (-1)^{i-1}\frac{\hbar}{m_0}k_z p + \frac{\hbar^2 k_z^2}{2m_l} + \frac{\hbar^2 k_x^2}{2m_t} + \frac{\hbar^2 k_y^2}{2m_t}, \quad (1)$$

where m_0 is the free electron mass, m_t is the transversal, and m_l is the longitudinal effective mass. The values of k_z are counted from the X point and are thus negative. The coupling between the two bands is described by the off-diagonal terms [4]:

$$H^0_{12}(k) = \frac{\hbar^2 k_x k_y}{M}. \quad (2)$$

The parameter M can be obtained from **k·p** perturbation theory:

$$\frac{1}{M} = \frac{1}{m_0}\left|\sum_{l\neq 1,2}(p_y)_{1l}(p_z)_{l2}\left(\frac{1}{E_l - E_1} + \frac{1}{E_l - E_2}\right)\right|.$$

We use for M the value computed by the empirical pseudo-potential method (EPM) at the $k_z = -k_0$ point, where the numerical value is close (but not equal) to $M \approx m_t/(1 - m_t/m_0)$ reported in [4]. With degenerate perturbation theory we obtain the following dispersion relation:

$$E(\mathbf{k}) = \frac{\hbar^2 k_z^2}{2m_l} + \frac{\hbar^2(k_x^2 + k_y^2)}{2m_t} - \sqrt{\left(\frac{\hbar}{m_0} k_z p\right)^2 + \left(\frac{\hbar^2 k_x k_y}{M}\right)^2}. \quad (3)$$

Expanding (3) around the minimum $k_z = -k_0$ with respect to $p_z = k_z - k_0$, one obtains a simplified dispersion relation [5]:

$$E(\mathbf{k}) = \frac{\hbar^2 p_z^2}{2m_l} + \frac{\hbar^2(k_x^2 + k_y^2)}{2m_t} - \frac{\left(\hbar^2 k_x k_y\right)^2}{M^2 \Delta} - \frac{\Delta}{4}, \quad (4)$$

where $\Delta = 2\hbar k_0 p/m_0 = 2\hbar^2 k_0^2/m_l$ is the gap between the Δ_1 and the $\Delta_{2'}$ conduction bands at $k_z = -k_0$. To estimate the value of the non-parabolicity parameter, we follow [5], Appendix B, and average out the angular dependence in (4). Assuming $\alpha E(\mathbf{k})$ to be small and recasting terms, one finally obtains:

$$\frac{\hbar^2(k_x^2 + k_y^2)}{2m_t} = E(\mathbf{k})\left(1 + \alpha E(\mathbf{k})\right),$$

with

$$\alpha = \frac{1}{2\Delta}\left(\frac{m_t}{M}\right)^2. \quad (5)$$

Substituting Si parameter values in (5), we estimate $\alpha = 0.6$ eV^{-1}, which is close to the phenomenological value of $\alpha = 0.5$ eV^{-1}.

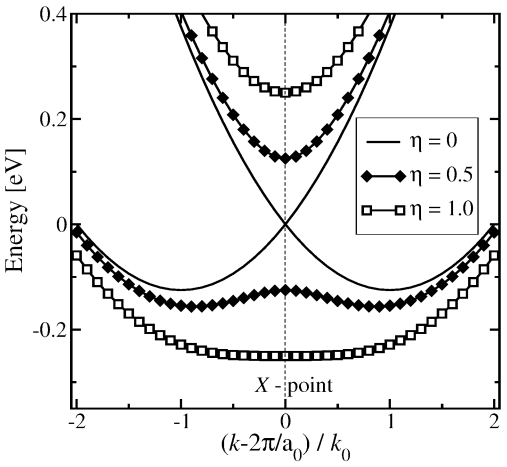


Figure 1: Conduction band profile along [001] for different strain values in [110] direction.

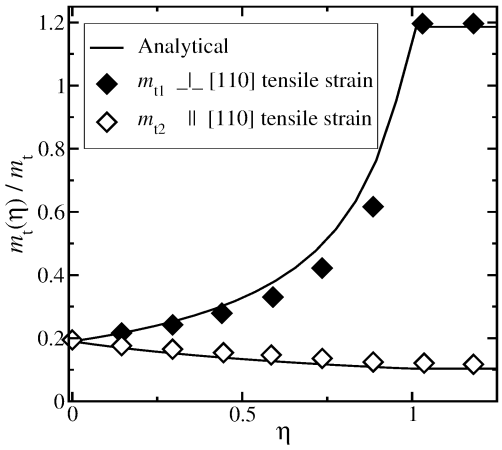


Figure 2: Dependences of the transversal effective mass on shear stress. The mass develops two branches.

CONDUCTION BAND MODIFICATIONS UNDER SHEAR STRAIN

Uniaxial stress along the [110] direction generates diagonal (ε_{ii}) as well as off-diagonal (ε_{xy}) components of the strain tensor in the principal coordinate system. The diagonal components are added to the diagonal matrix elements (1) of the [001] valley [6]:

$$H_{ii} = H_{ii}^0 + \delta E_C, \quad (6)$$

where $\delta E_C = \Xi_d\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + \Xi_u\,\varepsilon_{zz}$, with Ξ_d denoting the dilation and Ξ_u the uniaxial deformation potentials for the conduction band. The off-diagonal elements of the Hamiltonian are modified by the shear strain components [4]:

$$H_{ij}(k) = H_{ij}^0 - D\varepsilon_{xy}, \quad (7)$$

where $D = 14$ eV denotes the shear deformation potential.

The dispersion relation of the [001] valleys including the shear strain component for the conduction band now reads as:

$$E(\mathbf{k}) = \frac{\hbar^2 k_z^2}{2m_l} + \frac{\hbar^2(k_x^2 + k_y^2)}{2m_t} + \delta E_C - \sqrt{\left(\frac{\hbar}{m_0} k_z p\right)^2 + \left(D\varepsilon_{xy} - \frac{\hbar^2 k_x k_y}{M}\right)^2}. \quad (8)$$

Valley shifts

The off-diagonal strain component lifts the degeneracy between the two lowest conduction bands at the X points along the [001] axis. Since the conduction band minimum along the [001] axis is located near the

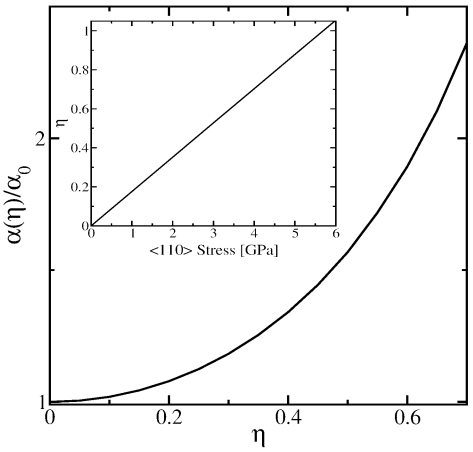


Figure 3: Analytical expression for dependence of the non-parabolicity parameter on stress. Inset: relation between normalized stress and stress in GPa.

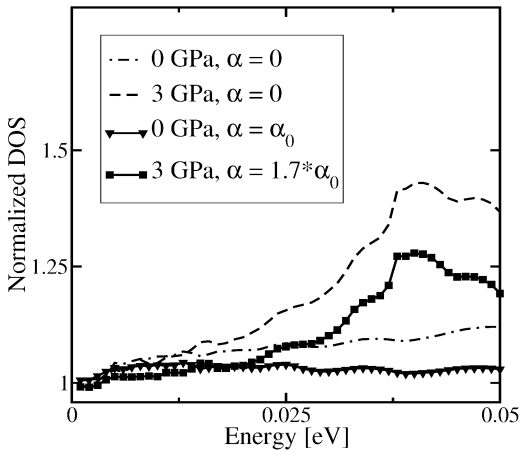


Figure 4: Numerical DOS in [001] valley normalized to the DOS analytical expressions obtained without (dotted lines) and with stress dependent nonarabolicity parameter (solid lines), for unstrained Si and Si under [110] uniaxial stress.

X point, the gap at the X point affects the position of the minimum as illustrated in Fig. 1. The minimum $k_{\min}$ moves closer to the X point. From (8) we obtain

$$k_{\min} = -k_0\sqrt{1-\eta^2}, \quad |\eta| < 1 \tag{9}$$

Here, the dimensionless off-diagonal strain $\eta = 2D\varepsilon_{xy}/\Delta$ is introduced. For $\eta \geq 1$ the valley minimum stays exactly at the X point.
The minimum of the [001] valley moves down in energy with respect to the remaining four fold degenerate valleys. For $\eta \leq 1$ the strain dependence is quadratic, while it is linear for $\eta \geq 1$:

$$\Delta E_{\text{shear}} = \begin{cases} -\dfrac{\Delta}{4}\eta^2 & , \quad |\eta| < 1 \\ -\dfrac{\Delta}{4}(2|\eta| - 1) & , \quad |\eta| > 1 \end{cases} \tag{10}$$

The shifts predicted by (10) were compared with results from EPM calculations and excellent agreement was found.

Stress-dependent effective masses

A shear strain component ε_{xy} modifies the effective masses in the [001] valleys. Evaluating the second derivatives of (8) at the band minimum (13), we obtain two different branches for the effective mass across (m_{t1}) and along (m_{t2}) the stress direction [110]:

$$\frac{m_t}{m_{t1}(\eta)} = \begin{cases} \left(1 - \eta\dfrac{m_t}{M}\right) & , \quad |\eta| < 1 \\ \left(1 - \text{sgn}(\eta)\dfrac{m_t}{M}\right) & , \quad |\eta| > 1 \end{cases} \tag{11}$$

$$\frac{m_t}{m_{t2}(\eta)} = \begin{cases} \left(1 + \eta\dfrac{m_t}{M}\right) & , \quad |\eta| < 1 \\ \left(1 + \text{sgn}(\eta)\dfrac{m_t}{M}\right) & , \quad |\eta| > 1 \end{cases} \tag{12}$$

Here, $\text{sgn}(\eta)$ denotes the sign function. In Fig. 2 the analytical expressions for the transversal masses (11) and (12) are compared with the masses obtained from EPM calculations. To improve the agreement at high values of stress $\eta \approx 1$, the deformation potential D is set to be slightly stress dependent in the form $D(\eta) = D + \beta\eta^2$, with $\beta = 0.7$ eV.

Stress and non-parabolicity

Shear strain affects the value of the non-parabolicity parameter α as well. In order to find the dependence, we rewrite the dispersion relation (8) in the vicinity of the valley minimum $k_{\min}$, which in the case of $k_x, k_y \neq 0$ is equal to

$$k_{\min} = -k_0\sqrt{1 - 4\left(\frac{D\varepsilon_{xy} - \frac{\hbar^2 k_x k_y}{M}}{\Delta}\right)^2}, \tag{13}$$

and, therefore, depends on $k_x k_y$. Expanding (8) around the minimum (13) for small $p_z = k_z - k_{\min}$, one obtains:

$$\begin{aligned} E(\mathbf{k}) = \; & \frac{\hbar^2 p_z^2}{2m_l(\eta)} + \frac{\hbar^2(k_x^2 + k_y^2)}{2m_t} - \frac{\Delta}{4} \\ & - \frac{1}{\Delta}\left(D\varepsilon_{xy} - \frac{\hbar^2 k_x k_y}{M}\right)^2, \end{aligned} \tag{14}$$

Taking into account the shifts of the valley minima (13) and the effective mass changes (11), (12), we rewrite

(14) as

$$E(\mathbf{k}) = \frac{\hbar^2 p_z^2}{2m_l(\eta)} + \frac{\hbar^2 \tilde{k}_x^2}{2m_{t2}(\eta)} + \frac{\hbar^2 \tilde{k}_y^2}{2m_{t1}(\eta)} - \frac{\hbar^4 \left(\tilde{k}_x^2 - \tilde{k}_y^2\right)^2}{2M^2\Delta}, \quad (15)$$

where $\tilde{k}_x = (k_x + k_y)/\sqrt{(2)}$ and $\tilde{k}_y = (k_x - k_y)/\sqrt{(2)}$ are the momentum projections in the rotated coordinate system, where the new x axis is along the [110] stress direction. The non-parabolic term written in the original xy coordinate system is similar to the corresponding term in (4). However, the parabolic term contains the effective masses $m_{t1}(\eta), m_{t2}(\eta)$, which are modified due to [110] stress. Introducing new variables $\tilde{x} = p_x/\sqrt{m_{t2}(\eta)}, \tilde{y} = p_y/\sqrt{m_{t1}(\eta)}$, we separate the renormalization due to stress of the parabolic part of the density-of-states (DOS) from the non-parabolic contribution. One rewrites (4) at $k_z = -k_0(\eta)$ as

$$E(\tilde{x}, \tilde{y}) = \frac{\hbar^2 \left(\tilde{x}^2 + \tilde{y}^2\right)}{2} - \left(m_{t2}(\eta)\tilde{x}^2 - m_{t1}(\eta)\tilde{y}^2\right)^2 \frac{\hbar^4}{2M^2\Delta}. \quad (16)$$

The last term gives the correction to the density of states of the parabolic bands. Assuming the last term to be small, we compute the correction to the DOS identifying it with the isotropic non-parabolicity parameter $\alpha(\eta)$. Taking into account the dependences (11) and (12) of the masses $m_{t1}(\eta), m_{t2}(\eta)$ on the stress η, we arrive at an expression for the strain dependent non-parabolicity parameter $\alpha(\eta)$:

$$\alpha(\eta) = \alpha_0 \frac{1 + 2(\eta m_t/M)^2}{1 - (\eta m_t/M)^2} \quad (17)$$

Expression (17) is plotted in Fig. 3. The non-parabolicity parameter depends on the square of the stress value and therefore does not depend on whether stress is tensile or compressive. This is expected, since tensile stress in [110] direction produces compression along $[1\bar{1}1]$, thus α can not depend on the sign of stress. Due to the quadratic dependence, the relative increase of $\alpha(\eta)$ is more pronounced at relatively large stress values.

RESULTS AND DISCUSSION

Fig. 4 illustrates the density-of-states of the [001] valley as a function of energy relative to the minimum, obtained numerically from empirical pseudo-potential calculations, for the unstressed case and stress equal to 3 GPa. The value of the non-parabolicity parameter at 3GPa is 70% higher than the unstrained value α_0, according to (17). The DOS is normalized to the analytical DOS corresponding to the parabolic dispersion

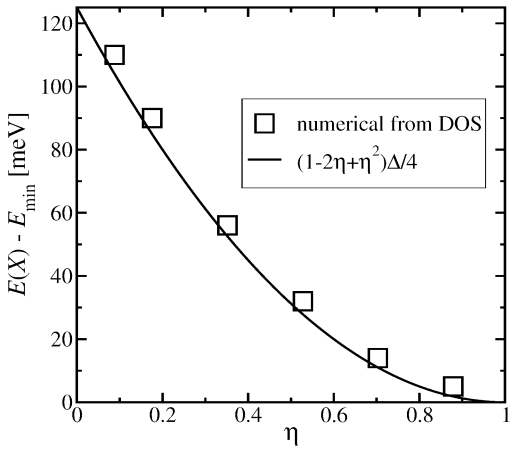


Figure 5: Energy at the X point relative to the value at the minimum. Solid line is a theoretical prediction based on (8).

(dotted lines), with strain dependent transversal effective masses (11), (12) and strain dependent longitudinal mass $m_l(\eta)$. The dependence of $m_l(\eta)$ on η is taken in the form [2, 3]:

$$m_l(\eta)/m_l = \begin{cases} (1-\eta^2)^{-1} & , \quad |\eta| < 1 \\ (1 - 1/|\eta|)^{-1} & , \quad |\eta| > 1 \end{cases} \quad (18)$$

Without the non-parabolicity parameter included, considerable deviations in the ratio of the numerical DOS to its analytical value is observed at higher energies. The ratio becomes much closer to unity, if the non-parabolicity correction due to stress dependent $\alpha(\eta)$ is taken into account in the analytical DOS (Fig. 4, solid lines). However, a deviation of the numerical DOS from the analytical model with the stress dependent non-parabolicity parameter is observed for energies larger than k_BT at high stress value. This happens due to the fact that the energy difference between the value at the minimum and the value at the X point decreases with stress (see Fig. 1). A pronounced peak in the numerical DOS appears, which corresponds to the flat dispersion close to X. The energy value of the peak is shown in Fig. 5 for different stress values and compared with theoretical predictions based on (8). For stress values larger than 3 GPa the energy difference from the minimum to the value at the X point becomes smaller than $2 \times k_BT$, as seen from Fig. 5, and a full-band description is required [2]. Through the modification of the density of the after-scattering states, the non-parabolicity parameter affects scattering rates and thus the mobility of the system. As example we consider the mobility in a double-gate FET sketched in Fig. 6. The subband energies and corresponding wave functions are calculated from the Schrödinger equation coupled self-consistently with the Poisson equation, for each value

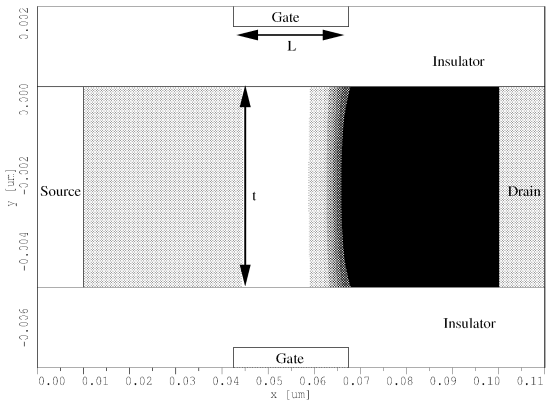


Figure 6: Sketch of a double-gate MOSFET structure. The contour plot shows the potential profile of the conduction band at the source-drain bias of 0.3 V and gate voltage 0.0 V.

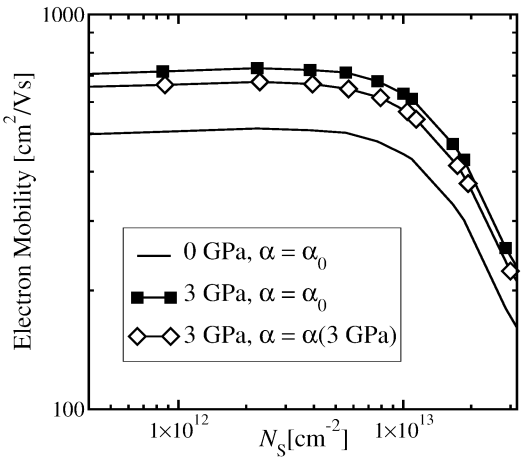


Figure 7: Mobility in a 3 nm thick SOI FET at 3 GPa [110] stress computed with and without stress dependence of the non-parabolicity parameter.

of the effective field. The wave functions are then used to calculate the scattering rates. Our transport calculations account for electron-phonon interactions [5, 7] and surface roughness scattering, which are the dominant mechanisms determining the mobility in the region of high effective fields. We use the original formulation by Prange and Nee [8, 9, 10] of the surface roughness scattering matrix elements.

Results of the mobility simulations in an ultra-thin body FET with strain along the [110] direction, with and without stress-dependence of the the non-parabolicity parameter taken into account, are shown in Fig. 7 together with the mobility in an unstrained SOI FET. Due to the stress dependence of the transversal mass the mobility the in UTB FET under uniaxial stress is significantly enhanced. However, when the stress dependence of the non-parabolicity parameter is included, the mobility enhancement is suppressed by almost 25% in a 3 nm thick SOI FET at a stress level of 3 GPa (Fig. 7).

CONCLUSION

The dependence of the non-parabolicity parameter on shear strain is analyzed for the first time. An analytical expression of the non-parabolicity parameter stress dependence is obtained and verified by comparing the corresponding DOS to the numerical DOS obtained with the empirical pseudo-potential method. The stress dependence of the non-parabolicity parameter results in almost 25% decrease of the mobility enhancement in a 3 nm thin SOI FET at high stress levels.

ACKNOWLEDGMENT

This work, as part of the European Science Foundation EUROCORES Programme FoNE, was supported by funds from FWF (project I79-N16), CNR, EPSRC and the EC Sixth Framework Programme, under Contract N. ERAS-CT-2003-980409, and by the Austrian Science Fund, project P19997-N14.

REFERENCES

[1] K. Uchida, T. Krishnamohan, K. C. Saraswat, and Y. Nishi, in *IEDM Techn. Dig.* (2005), pp. 129–132.

[2] E. Ungersboeck *et al.*, IEEE Trans.Electron Devices **54**, 2183 (2007).

[3] V. Sverdlov, E. Ungersboeck, H. Kosina, and S. Selberherr, in *Proc. EUROSOI 2007* (January, 2007), pp. 39–40.

[4] J. C. Hensel, H. Hasegawa, and M. Nakayama, Physical Review **138**, A225 (1965).

[5] C. Jacoboni and L. Reggiani, Reviews of Modern Physics **55**, 645 (1983).

[6] I. Balslev, Physical Review **143**, 636 (1966).

[7] P. J. Price, Ann.Phys. **133**, 217 (1981).

[8] R. E. Prange and T. W. Nee, Physical Review **168**, 779 (1968).

[9] M. V. Fischetti *et al.*, J.Appl.Phys. **94**, 1079 (2003).

[10] D. Esseni, IEEE Trans.Electron Devices **51**, 394 (2004).

PROCESS SIMULATION

MODELLING HARDWOOD DELIGNIFICATION IN CONTINUOUS DOWNFLOW LO-SOLIDS™ COOKING

Minna Palosaari and Timo Ahvenlampi
Systems Engineering Laboratory
P.O. Box 4300
90014 University of Oulu
Finland
E-mail: {minna.palosaari|timo.ahvenlampi}@oulu.fi

KEYWORDS
Modelling, Simulation, Delignification, Hardwood

ABSTRACT

In this paper, a new model for the simulation of hardwood delignification in Downflow Lo-Solids™ cooking is presented and investigated. The new simulation model is a modification of Gustafson's Kappa number model. The kraft pulping process and the delignification model development are shown in general. Industrial data is used in the validation of the proposed model and good results are achieved. The new simulation model is found to be more accurate in normal production and especially the recovery from a process disturbance is significantly faster. The results of the new Kappa number simulation model are shown together with the simulations of the earlier models. The differences between the measured and the modelled Kappa numbers are studied both qualitatively and quantitatively.

INTRODUCTION

Kraft pulping is the most important industrial method for producing chemical pulp; it is used in producing over 80 % of chemical pulps worldwide. It has the advantage of being insensitive to the wood type. All wood species can be used as raw material. Inorganic process chemicals can be efficiently recovered and reused in kraft processing. Kraft pulp has strong fibers and can be used in various paper products. The biggest problem in kraft pulping is the high amount of residual lignin. In chemical pulping, the lignin is dissolved from the wood chips to liberate the fibers. The process is called delignification and the progression is described typically with the Kappa number. (Gellerstedt *et al.* 2004)

This study concentrates on modelling Scandinavian birch pulping with Gustafson's (Gustafson 1983) model. Birch belongs to hardwoods, which have shorter fibers than softwoods. In the paper by Chiang *et al.* (1988), it was discovered that hardwoods consume more effective alkali than softwoods prior to the initial phase. The total consumption in the end of the initial delignification phase is the same for softwood and for hardwood. They came to a conclusion that this results from the reactive nature of the hardwood lignin. Hardwoods have a different chemical and physical structure than softwoods, so the chemical reactions and penetration of the liquids occur in a dissimilar way. The bulk delignification rate for hardwood pulping was found to be 1.2–1.8 times faster than for softwood pulping by Chiang *et al.* (1988). For these reasons, the hardwood pulping has to be modelled separately from the softwood pulping.

Gustafson *et al.* (1983) compared the models developed earlier for the delignification kinetics and generated their own theoretical model. The model is widely accepted to be the most reliable in describing the kinetics of the pulping. Yet, Vroom's (1957) H-factor and Hatton's (1973) Kappa number model are still commonly used in the industry. In the paper by Gustafson *et al.* (1983), the model was developed originally for softwoods. Rantanen (2006) combined Gustafson's kinetics with Härkönen's (1987) fluid dynamics and developed models for two different kraft pulping processes. He optimized the models' parameters separately to fit both softwood and hardwood pulping. This article introduces a modification of the optimized Gustafson's model, which recovers faster from disturbances and is also more accurate in normal operation. Only the delignification and Kappa number modelling are studied. The models are evaluated by comparing the Kappa number simulations and the differences between the modelled and measured Kappa numbers.

MODEL DEVELOPMENT

Rantanen (2006) used the rate equations from Gullichsen (2000) to describe the delignification in the pulping process. The equations are developed from the Gustafson's model (Gustafson *et al.* 1983) for Scandinavian pine sample that initially contains 27 % lignin, 67 % carbohydrates, and 6 % extractives and ash. For the initial delignification applies Equation (1),

$$\frac{dL}{dt} = k_{il} e^{(17.5-8760/T)} L \,, \qquad (1)$$

where k_{il} is a species specific reaction rate constant in the initial phase.

The bulk phase of the delignification starts when the lignin content is about 20 % of the wood, and the transition is

independent from temperature, sulfide, and alkali concentration (Olm and Tistad 1979). The rate equation for bulk delignification is shown in Equation (2) below (Gullichsen 2000),

$$\frac{dL}{dt} = k_{0bl} e^{(A_1 - B_1/T)} \left[OH^-\right] L + k_{1bl} e^{(A_2 - B_2/T)} \left[OH^-\right]^{0.5} \left[HS^-\right]^{0.4} L \quad (2)$$

where k_{0bl} and k_{1bl} are species specific reaction rate constants in the bulk phase.

For the residual delignification, Gullichsen (2000) suggests the Equation (3),

$$\frac{dL}{dt} = k_{rl} e^{(19.64 - 10804/T)} \left[OH^-\right]^{0.7} L, \quad (3)$$

where k_{rl} is a species specific reaction rate constant.

In the simulation framework, the reaction rate parameters are updated on-line based on the Kappa number measurements.

Rantanen (2006) optimized the parameters of the bulk phase equation (Eq. 2) for softwoods and hardwoods separately. The original and optimized parameters are shown in Table 1. The initial and residual phase delignification models are used with original parameters in the simulations. Both the originals and the optimized parameters for the bulk phase are used in the simulations.

Table 1: Original (Gullichsen 2000) and Optimized Parameters (Rantanen 2006)

Parameter	Original value, SW	Optimized value, SW	Optimized value, HW
A_1	35.5	29	28
B_1	17200	17000	16800
A_2	29.4	15.9	4.7
B_2	14400	8500	3800

NEW SIMULATION MODEL FOR HARDWOOD DELIGNIFICATION

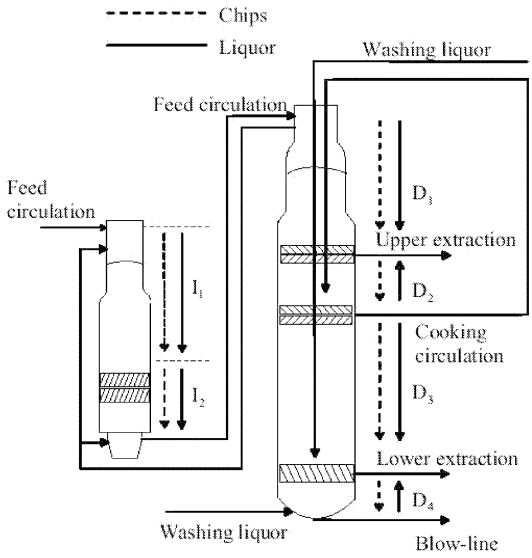
As discussed earlier, Gustafson (Gustafson *et al.* 1983) developed the model for softwoods. In the paper by Chiang *et al.* (1988), it was found that hardwood lignin is more reactive in nature than softwood lignin. This article is based on studies, which concentrate on adapting the Gustafson model for hardwoods. Rantanen (2006) optimized the parameters of the Arrhenius part of the model (Table 2), but the results were not satisfactory in hardwood modelling.

In the present study, Gustafson's model is developed further to describe hardwood delignification more accurately. Smaller differences between the modelled and the measured Kappa numbers are achieved by adding constants to the simulation model, which describes the delignification. Gustafson's model is sensitive to process disturbances, both wanted and unwanted. The effect on the Kappa number model is too strong especially in modelling hardwood. The constants make the model more stable and the recovery from the process disturbances faster. The new simulation model for hardwood delignification is executed as in Equation (4),

$$\frac{dL_{new}}{dt} = \frac{dL_{original}}{dt} \cdot A + B, \quad (4)$$

where L_{new} is the new lignin yield, $L_{original}$ is the modelled lignin yield from Equations 1–3, A is the percentage of the modelled lignin yield, and B is the constant lignin yield.

In this study, the data is from the industrial Downflow Lo-Solids™ cooking process (Fig. 1). The data has been collected from a plant's automation system.



Figures 1: Simplified Illustration of the Downflow Lo-Solids™ Cooking Process

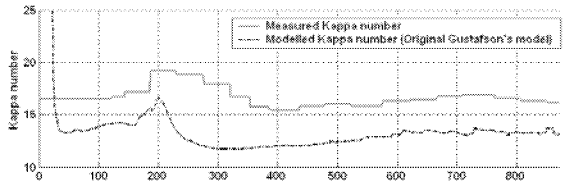
Equation 4 is applied to the initial and bulk phases in real-time simulations. Due to the small amount of delignification, the residual phase is simulated with Gustafson's model without any constant lignin yield. The parameters are experimentally determined; see Table 2. The model is found to give the best fitting when 50 % of the modelled lignin yield is considered. The process is divided into eight calculation zones.

Table 2: Simulation Parameters A and B

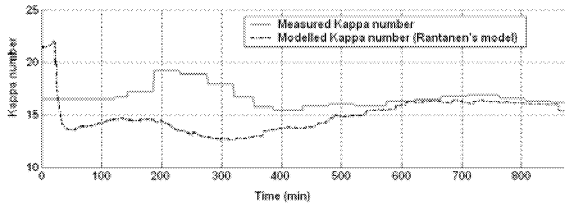
Calculation zone	A	B
Initial I_1	0.5	2.8
Initial I_2	0.5	0.7
Bulk D_1	0.5	0.9
Bulk D_2	0.5	1.3
D_{3a}	0.5	1.8
D_{3b}	0.5	1.4
D_{3c}	0.5	1
D_{3d}	0.5	0.7

RESULTS

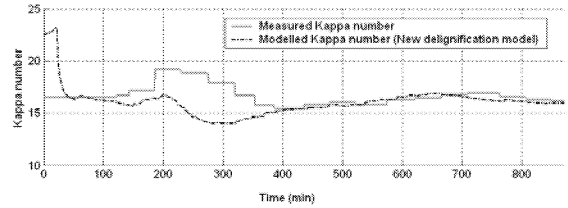
The real-time Kappa number models are simulated and the models are analyzed qualitatively (visual observation) and quantitatively by the root mean squared error calculations. The modelling and simulations are done using Matlab®. In the beginning of the simulation period, shown in Figures 2–4, the process is recovering from a disturbance.



Figures 2: The Modelled and Measured Kappa Numbers with the Original Gustafson's Parameters



Figures 3: The Modelled and Measured Kappa Numbers with the Optimized Parameters by Rantanen (2006).



Figures 4: The Modelled and Measured Kappa Numbers with the Proposed Simulation Model.

The calculated root mean squared differences between the modelled and the measured Kappa numbers are shown in Table 3. The RMSE values are first calculated for a simulation period with normal operation and second for the simulation period shown above in which the process is recovering from a disturbance.

Table 3: RMSE–values between the Modelled and Measured Kappa numbers

Normal operation, Original model	0.6343
Normal operation, Rantanen's model	0.2955
Normal operation, New simulation model	0.1855
Disturbed production, Original (Fig.2)	4.1687
Disturbed production, Rantanen (Fig. 3)	2.7490
Disturbed production, New model (Fig. 4)	1.8894

DISCUSSION AND CONCLUSIONS

From Figures 2–4 and RMSE calculations (Table 3), it can be seen that the modelling has progressed. From the simulations, a lower difference between the modelled and measured Kappa numbers can be visually observed. The RMSE calculations give a numeric value for the differences. The RMSE is the lowest in both of the studied simulation cases with the new simulation model. The progress in the modelling is shown from the original Gustafson's model to a proposed simulation model for hardwood pulping. The new simulation model is validated using existing models as reference results.

Rantanen (2006) optimized the parameters of the Gustafson's model for hardwoods, and this article presents a modification, which is more accurate in steady production, and the recovery after a disturbance is faster with the modified delignification model. The disturbances are common in real processes and those should be acknowledged in the modelling. Gustafson's delignification model describes the delignification phenomenon well, but it needs some adapting when it is applied to a real process. The necessary adapting is conducted as in Equation 4 in this study. The parameters in Table 2 are developed for a Downflow Lo-Solids cooking process. The parameters are process-specific. Also the applicability of the proposed method for other cooking plants is studied. A good simulation model is necessary in predicting and controlling the behaviour of a real cooking process.

REFERENCES

Chiang, V.L; R.J. Puumala; H. Takeuchi; and R.E. Eckert. 1988. ''Comparison of softwood and hardwood kraft pulping''. *Tappi Journal* 71, No.9, 173–176.

Gellerstedt G.; A. Majtnerova; and L. Zhang. 2004. ''Towards a new concept of lignin condensation in kraft pulping: Initial results''. *Comptes Rendus Biologies* 327, 817–826.

Gullichsen J. 2000. ''Chemical engineering principles of fiber line operations''. In *Chemical Pulping.* Papermaking Science and Technology, Book 6A, J. Gullichsen and C-J. Fogelholm (Eds.). Fapet Oy, Jyväskylä, Finland, 244–327.

Gustafson R.R.; C.A Sleicher; W.T. McKean; and B.A. Finlayson. 1983. ''Theoretical model of the kraft pulping process''. *Industrial & Engineering Chemistry Process Design and Development* 22, No.1, 87–96.

Hatton J.V. 1973. ''Development of yield prediction equations in kraft pulping.'' *Tappi Journal* 56, No.7, 97–100.

Härkönen E.J. 1987. ''A mathematical model for two-phase flow in a continuous digester''. *Tappi Journal* 70, No. 129, 122–126.

Olm L. and G. Tistad. 1979 ''Kinetics of the initial stage of kraft pulping''. *Svensk Papperstidning* 82, No. 15, 458–464.

Rantanen R. 2006. *Modelling and Control of Cooking Degree in Conventional and Modified Continuous Pulping Processes*, PhD Thesis, Department of Process and Environmental Engineering, University of Oulu.

Vroom K.E. 1957 ''The H-Factor: A means of expressing cooking times and temperatures as a single variable.'' *Pulp and Paper Magazine* 58, No.3, 228–231.

DATA ANALYSIS AND SIMULATION FOR QUEUEING SYSTEMS

Luigi Rarità
Eliza Trapel
Department of Information Engineering and Applied Mathematics
University of Salerno
84084, Fisciano (Salerno), Italy
E-mail: {rarita, eliza}@diima.unisa.it

KEYWORDS
Queueing systems of general type, discrete – event simulation, Mathematica

ABSTRACT

A numerical package for the simulation of general queueing systems, implemented with Mathematica, is described. The realized tool presents a graphic user – friendly interface and a section for data analysis, that can be useful in order to individuate the input of a queueing model. The tool is used to study and optimize a real queueing system, which is the post office placed in Fisciano, Italy.

INTRODUCTION

Real world applications are often so complex that it is impossible a complete description with analytical results. For this reason, simulative procedures are often used.
In this paper, we want to present a simulative tool, able to analyze the behaviour of queueing systems. Particularly, this tool is based on discrete – event simulation and can be useful in order to analyze queueing systems with general interarrival and service times, for which very few analytical results are available. Numerical tools for the simulation of queueing systems are already available. The simulative tool, that we present, is able not only to simulate general queueing systems in asymptotic regime, but also in transient one, and allows a graphical representation of the obtained results. It uses the numerical and graphical power of Mathematica. Moreover, such tool, characterized by a simple user – friendly interface, starting from the numerical obtained results, is able to suggest the user a policy of optimization for the queueing system object of study.
A preliminary version of the tool was already presented in (Cascone et alii, 2005). The version, that we consider in this paper, is enriched by the introduction of a step by step guide for data analysis of real phenomena involved in queueing systems study (see Fishman, 1973) and by the transient analysis. The introduction of such guide is highly
not trivial, since most natural phenomena, that can be modelled by queueing systems, are analyzed starting from data samples.
In order to have a better comprehension of the potentialities of the simulative tool, we present the simulation of a real queueing system constituted by the post office of Fisciano, Italy. In particular, for such institute, in a working week, interarrival times and service ones for users were kept. The data analysis, made through such samples, allowed to establish that the phenomena connected to service discipline and arriving process of users are of general type. For the study of performance indices, it is necessary a simulative procedure. After that a complete study of data samples and performance indices was made, an optimization policy for the queueing system under consideration is presented. A functional, that takes into account various types of costs weighted by opportune coefficients, for the optimization is defined, and used to underline an hypothetical optimization policy, that the post office could use during days in which it is interested by a strong flow of customers. In particular, the use of the simulative tool shows how, starting from a defined number of servers, the post office could improve its own quality of service adding other servers, respecting constraints on the allowed money budget.
The paper is organized as follows: first, we present the simulative tool realized by Mathematica; then, we consider the study of the performance indices and the optimization for the post office described before.

SIMULATIVE TOOL

Simulative algorithm

In order to simulate queueing systems, it is necessary to generate random variables. The distributions, that have to be generated, clearly depend on the characteristics of the system. For example, for a $M/M/1$ queueing system, random variates from an exponential distribution are necessary. The Mathematica code for the generation starts by a random – number generator, that is invoked to generate a uniform random variable between 0 and 1. The wished distributions are obtained by implementation of some opportune transformations.
For the simulation of queueing systems, two only types of events are necessary: the arrival of a customer to the system; the departure of a customer from the system, after he has completed his own service. For a better comprehension of the events management, we can present a flowchart, that makes a summary of the tool activities (Figure 1):

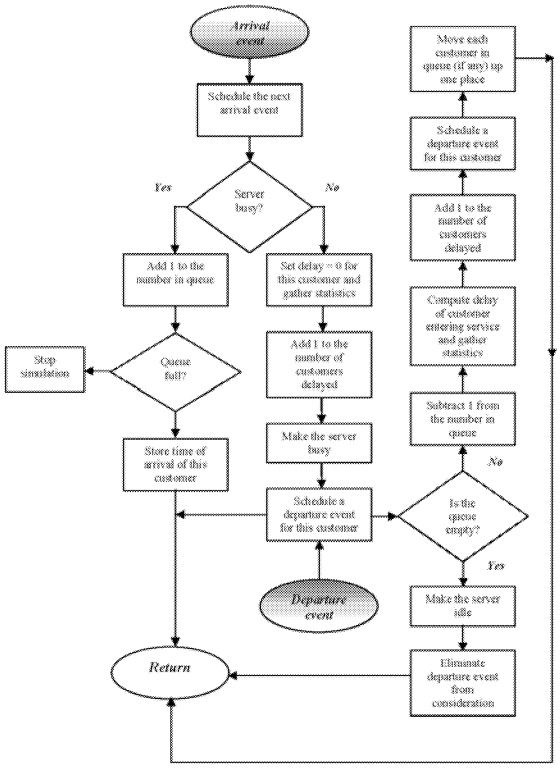


Figure 1: Flowchart for arrival and departure routine.

From Figure 1, it can be noted that the time of the next arrival in the future is generated and placed in the event list. Then, a check is made to understand if the server is busy. If the server is busy, the number of customers in queue is incremented by one and a test is made in order to establish if the queue is full. If the queue is not full, the arrival time of the new customer is put at the end of queue. Moreover, if the arriving customer notices that the server is idle, then he has a delay zero (a delay zero can be considered as a real delay), and the number of customer delays completed is incremented by one. The server becomes busy, and the time of departure from service of the arriving customer is placed in the event list.

Graphic interface

On the basis of the presented flowchart, a package, implemented in Mathematica, has been realized. Every operation of the implemented numerical tool can be made through some buttons that are contained in a Palette (Figure 2).
Now, we can make a simple example, that concerns the use of the Mathematica simulation tool. The first useful step, that the user must follow, is to click on the button "Loading the package". In this way, the tool is ready to execute the operations, that the user wishes, on an interface, that contains the Palette (Figure 2).

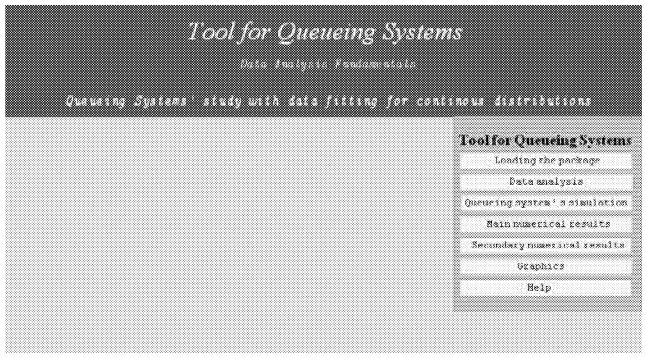


Figure 2: Palette and graphic work area.

The user is allowed to simulate queueing systems or make some data analysis on the graphic work area.
Let us consider customers arriving at a bar, where there is only one counter. Interarrival and service times of customers have been measured and so two data samples have been obtained. We can study the queueing system, that characterizes the bar, with the proposed tool. First of all, it is necessary to know what are the parameters of the distributions, to which data samples belong.
In order to analyze data samples, we have to push on the button "Data Analysis", which is in the Palette of the tool. So, a new "Data Analysis" button appears in the interface and the user, clicking on it, can upload the data, specifying the path of the two text files in which the collected data are stored. Now, the tool shows the results related to the following operations: the uploading of the two files, with an analysis of sample mean, variance, median, skewness and the histogram of the two data sets; makes tests (Chi Square and Kolmogorov – Smirnov tests) to understand which is the correct continuous distribution for interarrival times and service ones, and shows test results.
A series of buttons guides the user to understand which is the continuous distribution, that can be associated to the samples. In particular, Chi Square test (for which the user has to insert the number of intervals) and the Kolmogorov – Smirnov test, for the queueing system, that models the bar, do not reject the assumption that the interarrival times (related to the experiment we are making) are Weibull distributed, according to some levels of risk.
In particular, the data analysis has established that:

a) the interarrival times are Weibull distributed with a shape parameter $\alpha = 1.54562$ and $\beta = 3.07929$;
b) b) the service times are exponentially distributed with parameter $\mu = 0.52013$.

We show a table where we can see some results about the sample characteristics for the interarrival times of the queueing system, that we are considering:

Table 1: principal characteristics of data sample for interarrival times.

Minimum	0.0626634.
Maximum	9.71502
Mean	2.77127
Variance	3.33026

Median	2.46708
Skewness	0.964217

The histogram of the data sample, computed by the tool, is the following.

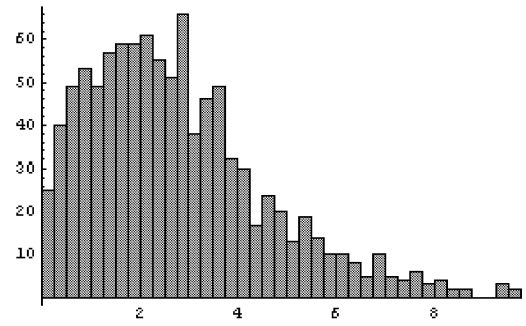


Figure 3: Histogram of the data sample for the interarrival times.

Moreover, the tool indicates the user that the behaviour of the histogram is justified through the sample computed characteristics. In fact, it is possible to observe that:

a) the mean is greater than the median, hence the real distribution is not symmetric;
b) the skewness is positive, hence the real data distribution has a right tail;
c) the variance has a value, that indicates a great variation of data around the mean value.

We report the tool results for the Chi Square test and the Kolmogorov Smirnov one for the interarrival times.
As for the Chi Square test, if the user chooses 13 degrees of freedom, we obtain that $\chi = 9.904$. The tool elaborates the picture of Chi Square distributions for 7, 9, 11, and 13 degrees of freedom (this last one in black in Figure 4).

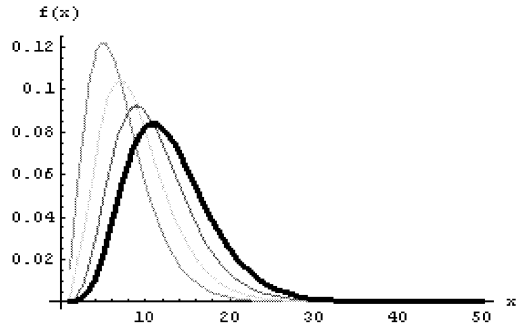


Figure 4: Pdf of Chi Square distributions.

The user can also understand the values of the quantiles of the distribution, related to 13 degrees of freedom. Such quantiles are in the following table, for different risk values.

Table 2: quantiles for Chi Square distribution.

Risk level (%)	***Quantile***
15	18.202
10	19.8119
5	22.362
1	27.6882

As $\chi = 9.904$ is less than the quantile values for every risk level, we can conclude that, for risks of 15%, 10%, 5%, and 1%, the Chi Square test is not rejected for the Weibull distribution.
As for the Kolmogorov Smirnov test, the results elaborated by the tool are the following: the test statistics is $D = 0.0187055$, while the correct statistics, computed with the assumption that the continous distribution is of Weibull type, is $KS = 0.59152$. The critic values of the correct Kolomogorov – Smirnov statistics are, for the distribution Weibull and different risk levels, in the following table:

Table 3: critic values for Kolmogorov – Smirnov test.

Risk level (%)	***Quantile***
10	0.803
5	0.874
2.5	0.939
1	0.944

As $KS = 0.59152$ is less than the quantile values for every risk level, we can conclude that, for risks of 10%, 5%, 2.5%, and 1%, the Kolmogorov – Smirnov test is not rejected for the Weibull distribution.
After collecting all the needed informations about interarrival and service times, it is possible to simulate a queueing system characterized by such distributions. Figure 5 shows the parameters, that we have to insert. First, we have to click on "Queueing system's simulation" button in the palette, and then we can choose the buttons "Simulation" and "Help" appear on the interface. When we push on "Simulation", we have to insert the parameters of the queueing system. In Figure 5, the user is simulating a queueing system with one server and a buffer of infinity capacity. It is very useful to follow all the instructions contained in the "Help" button for a correct inserting of parameters. Notice that, beside the complete informations about interarrival and service times, it is asked to specify the number of servers, the queue length, the dimension of the population, who asks for service, and the number of customers, who have come into the system (this last information is fundamental, as it establishes the clock of the simulation process).
When the simulation process ends, a notice appears in order to advise the user that it is possible to analyze the elaborated results. The user can choose to display only the main numerical results or all the other ones. He can also display the main results in a graphic form, clicking on "Graphics" on the Palette. A detailed Help button suggests the user which picture he can see. In Figure 6, the graphic, associated to the users in queue during the simulation steps, is presented. In this picture, we can see the time dependence of the simulated number of users in queue. In the table below, we report the results for the queueing system, whose characteristics were explained before. We indicate by N the average number of

users in the system, by N_Q the average users in the queue, by W the average waiting time in queue, by T the average waiting time in the system, by p the loss probability for the system.

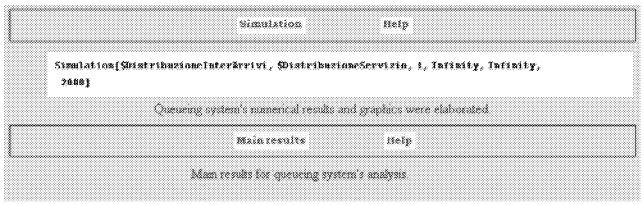


Figure 5: Simulation buttons.

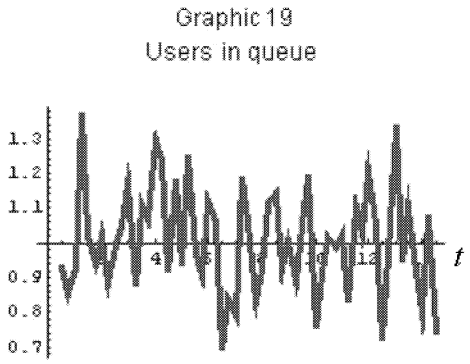


Figure 6: Number of users in queue.

Table 4: performance indices.

N	1.697
N_Q	1.003
W	2.784
T	4.712
p	0

APPLICATION

In this section, we consider an application of queueing systems, through which it is possible to consider the potentialities of the simulative tool elaborated by Mathematica. In particular, we refer to a real case study, the post office of Fisciano, Italy. For such institute, interarrival and service times of the users were kept. The data collection, useful in order to study the statistical behaviour of queues and servers of the institute, was made during a working week, in May 2007, from 9:00 a.m. to 2:00 p.m., from Monday 21 up to Friday 25. The post office, that is under consideration, is a multi – server queueing system, characterized by three servers, that serve the users according to a FCFS (first come, first served) discipline. Obviously, such three servers do not work together every day. Particularly, during the week in which data were collected, the three servers worked simultaneously only on Monday and Thursday. As for the other days of the week, only two servers were active and not always for the entire period of work.

The interarrival and service times analysis, that was made by the collected data, shows that the interarrival times and service ones are of general type. Particularly, for every day of the week, the interarrival times follow a Weibull distribution, while the service ones are Gamma distributed. The highest customers flow was on Thursday.

We show the main performance indices on Thursday (see the table below), obtained through simulations. In the table, we can compare the results obtained with a different number of servers.

Table 5: performance indices.

	1 server	*2 servers*	*3 servers*
N	4.42435	1.02931	0.833166
N_Q	3.66554	0.234996	0.0340397
W	3.29125	0.203997	0.0293978
T	3.99502	0.901372	0.733926
p	0.02282	$1.0\cdot10^{-3}$	$3.0\cdot10^{-5}$

Moreover, the interarrival times are Weibull distributed with parameters $\alpha = 0.92$ and $\beta = 0.85$; the service times are Gamma distributed with parameters $\alpha = 0.47$ and $\beta = 1.5$.

It is possible to note how, through comparison of simulative results, when three servers are active, the customers are served in a very fast way. From this, we could conclude that the post office should always use three servers. However, this approach cannot be very useful if we consider the costs, that the post office has to support.

Hence, it is possible to consider the results elaborated by the tool in order to optimize the system performances, taking into account various factors, among which costs.

We can introduce an objective function, J, under the assumption that it depends on the average service time, θ, the average waiting time in queue, $w(\theta)$, and the number of added servers, η. More precisely, the cost associated with providing the service increases as the level of service becomes better; the cost related to the average sojourn time of a customer in queue, that estimates customer satisfaction at the service level θ, decreases as the level of service is improved; hence, J increases when θ and $w(\theta)$ decrease and, naturally, grows when the number of added servers, η, becomes higher. We assume the following cost functionals:

$$J(\theta,\eta,\nu) = \frac{\alpha}{\theta} + \beta(1-0.1\eta)w(\theta) + \gamma\eta + \delta(\nu+\eta)^2 +$$

$$-\varepsilon\left[\log\left(\nu^2+\eta^2+1\right)\right],$$

where α,β,γ are weights belonging to $]0,1[$. Note that $\nu+\eta$ indicates the total number of active servers in the system when the starting number of working servers is ν. The natural logarithmic term $\log\left(\nu^2+\eta^2+1\right)$ corrects the quadratic dependence associated to the cost of the number of

active servers (such dependence is multiplied by δ, that indicates the average money, which each server receives), under the assumption that the post office had subsidies, indicated by ε. With $\eta \in \{0,1,2,3\}$, and $\nu \in \{1,2,3\}$, we get the cost function for: $\alpha = \beta = \gamma = 0.5$, $\delta = 1250$ Euros, and $\varepsilon = 650$ Euros, with an average waiting time $\theta = 1$ min. The picture of the cost functional *J* refers to the Thursday situation.

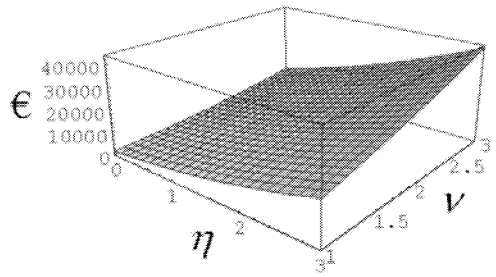


Figure 7: Cost functional *J*.

The behaviour of the cost functional indicates an average monthly situation based on input and service times statics we usually get on Thursday, the most crowded day. Hence, with the assumption that $\theta = 1$ min, we want to understand what is the cost obtained by variations of the number of added servers, once fixed the starting number of servers at a value $\nu \in \{1,2,3\}$. The following table reports the monthly variation of costs when new servers are added to the old ones.

Table 6: costs associated to added servers.

	1 starting server	***2 starting servers***	***3 starting servers***
0 added server	821.0292	3998.3475	9815.130
1 added server	4317.7389	10134.521	18505.687
2 added servers	10134.64	18631.231	29605.086
3 added servers	18505.937	29605.211	43164.348

If we suppose that the service activities costs, which the post office have to support, cannot exceed 20000 Euros, we can try to design a policy according to the values indicated in the previous table. For instance, if we have only one starting server, we could add other three servers. However, the most advantageous situation, from an economic point of view, is when the starting servers are three. In this case, respecting the constraint on the post office budget, we could add another sever. We can notice that the policy "three starting server" is the same adopted by the post office on every Thursday.

CONCLUSIONS

In this paper, we have presented a simulative tool implemented by Mathematica. Such tool is able to simulate general queueing systems, also considering real data connected to the phenomena, that describe queueing models, and treating some important aspects of queueing system study, such as time dependence performance indices.
A real application, constituted by a post office, was presented. The results obtained through the tool were useful to suggest an optimization policy for the post office.

REFERENCES

Law Averill M., Kelton W. David, "Simulation Modeling & Analysis", Second Edition, Mc Graw Hill, 1991.

Banks J., "Handbook of Simulation", John Wiley & Sons, 1998.

Banks J., Carson J.S., "Discrete Event System Simulation", Prentice Hall, Englewood Cliffs, 1984.

Bocharov P. P., D'Apice C., Pechinkin A.V., Salerno S., "Queueing Theory", Modern Probability and Statistics, VSP, The Netherlands, 2004.

Cascone A., Rarità L., Trapel E., "Simulation and Analysis of a bank's multi – server queueing system", *Transactions of XXV International Seminar on Stability Problems for Stochastic Models, XXV International Seminar on Stability Problems for Stochastic Models*, 20-24 September 2005, 20th – 24th, pages 89-95.

Fishman G. S., "Statistical Analysis for Queueing Simulations", *Management Science*, Vol. 20, No. 3, Theory Series, November 1973, pages 363-369.

Kleinrock L., "Queueing Systems", Vol. I, John Wiley & Sons, 1975.

BIOGRAPHY

LUIGI RARITÁ was born in Salerno, Italy, in 1981. He graduated cum laude in Electronic Engineering in 2004, with a thesis on mathematical models for telecommunication networks modelled by tandem queueing networks with negative customers and blocking. He is actually a PhD student in Information Engineering at the University of Salerno. His scientific interests are about queueing theory, self – similar processes, and fluid – dynamic models for traffic flows on road and telecommunication networks.
His e-mail address is `rarita@diima.unisa.it`.

ELIZA TRAPEL was born in Iassy, Romania, in 1980. She graduated in Mathematics and Physics in 2003, at the University "Alexandru Ioan Cuza, Iassy, Romania, with a thesis, that concerns the behaviour of convergence in topological spaces. She is actually a PhD student in Mathematics at the University of Salerno. Her scientific interests are about queueing theory and control theory.
Her e-mail address is `eliza@diima.unisa.it`.

Numerical Simulation of Stereolithographic Processes

João M. Matias
Paulo J. Bártolo

Centre for Rapid and Sustainable Product Development, Instituto Politécnico de Leiria
INDEA (Campus 5) Rua das Olhalvas
Leiria, Portugal
E-mail: {joao.matias|pbartolo}@estg.ipleiria.pt

ABSTRACT

Stereolithography is an important rapid prototyping process that creates three-dimensional solid objects in a multi-layer procedure. This technology involves the curing or solidification of a liquid photo or thermo sensitive polymer through the use of an irradiation light source, which supplies the energy needed to induce a chemical reaction, bonding large numbers of small molecules and forming a highly cross-linked polymer. This reaction determines the resin morphology that, in turn, defines the physical, electrical, and mechanical properties of the cured material. The cure reaction is highly exothermic and, the temperature and the reaction rate can vary considerably within the curing material due to polymers low thermal conductivity.

Simulations of stereolithography require analytical models describing accurately the curing kinetics. This paper proposes the use of appropriate irradiation, energetic and kinetic models to study photo-initiated curing reactions in stereolithographic applications. The proposed models are both theoretically rigorous and practical in its implementation, consisting of the resolution of a set of partial differential equations accurately describing the heat transfer mechanism, irradiation conditions and chemical reactions involved. These models are integrated and solved using the finite element method. The proposed simulation tool includes effects of initiator concentration, temperature and light intensity, to predict the diffusion-controlled effects occuring after vitrification, the phenomenon of unimolecular termination.

1. Introduction

Stereolithography is the most important rapid prototyping technology currently available. This technology involves the curing or solidification of a liquid photosensitive polymer through the use of an irradiation light source, which supplies the energy needed to induce a chemical reaction, bonding large numbers of small molecules and forming a highly cross-linked polymer. Typically, the stereolithographic apparatus consists of a computer, a vat containing a photosensitive polymer, a moveable platform on which the model is built, a laser to irradiate and cure the polymer, and a dynamic mirror system to direct the laser beam. The computer uses a sliced model information to control the mirrors, which direct the laser beam over the polymer surface, "writing" the cross-section of one slice of the model, by polymerising a set of elementary volumes called voxels. After drawing a layer, the platform dips into the polymer vat, leaving a thin film, from which the next layer is created. The next layer is drawn after a waiting period to re-coat the surface of the previous layer. The process is repeated, with each new layer adhering to the previous one. Finally, a post-cure operation is needed to complete the curing process.

The use of stereolithography is still in its infancy. Much of the technology and operating procedures are based on empirical correlations and work experiences and very little is known about the physical and chemical changes occurring in the material owing to the irradiation. A good understanding of the curing process is an important factor to improve both the precision and quality of the models, and to develop well-adapted polymeric systems. This paper proposes a novel computational tool using the finite element method to study, simulate and optimize stereolithography. The model, which is theoretically rigorous and practical in its implementation, describes both the heat transfer effects and the chemical reaction course.

2. Computational modelling

2.1 *Modelling the heat generation during the curing process*

The curing process is an exothermic reaction whose heat generation rate is important to be evaluated. The heat generation rate is described by:

$$\dot{Q} = \Delta H_{TOTAL} \times R_P \quad (1)$$

where ΔH_{TOTAL} is the total enthalpy variation and R_P is the reaction rate.

Three different approaches were developed to model the reaction rate:

- Energetic models;
- Mechanistic and semi-mechanistic models;
- Phenomenological models.

Energetic models

The energetic models were proposed by Yamaguchi and Nakamoto (1998) assuming that the cure process begins only when a critical value of energy is reached, which depends on the material being used. These models are based on a direct relation between the radiation intensity, radiation profile and energy, developed to simulate both direct irradiation and mask irradiation processes.

Mechanistic and semi-mechanistic models

Mechanistic models are based on both the concept of free radical polymerisation and the mechanism of reactions with diffusion (Yousefi et at, 1997; Bártolo, 2006). This approach requires several assumptions and approximations to simplify the complexity of the curing reaction. The rate constants in the free radical polymerisation reactions have been modified by introducing the molecular parameters of free volume and glass transition temperature incorporating the effect of diffusion. This way these models have been pointed out, the model equations often include many parameters, which must be determined using numerical optimization schemes.

Photo-curing processes are chain reactions characterized by three main steps: photo-initiation, propagation and termination. The photo-initiation mechanism occurs when photons of a laser radiation source collide with a photo-initiator molecule, generating the formation of reactive species. The addition of the reactive species to monomers, followed by successive additions forming a growing polymer chain comprises the propagation step. This propagation will continue until some reaction occurs, which terminates it.

A typical radical polymerisation mechanisms can be described as follows (Odian, 1981; Painter and Coleman, 1997):

$$\left.\begin{aligned} PI &\xrightarrow{h\nu} R\bullet \\ M + R\bullet &\xrightarrow{k_i} P_1\bullet \end{aligned}\right\} \quad \text{initiation}$$

$$P_n\bullet + M \xrightarrow{k_p} P_{n+1}\bullet \quad \text{propagation}$$

$$P_n\bullet + P_m\bullet \xrightarrow{k_{tc}} M_{n+m} \quad \text{termination by combination} \qquad (2)$$

$$P_n\bullet + P_m\bullet \xrightarrow{k_{td}} M_n + M_m \quad \text{termination by disproportionality}$$

where PI and M are respectively the photo-initiator and the monomer, R• is the primary radical, P_n• is the polymeric radical with a chain size of n units of monomer.

The rate of monomer depletion, called rate of polymerisation, is given by the following equation (Odian, 1981; Rodrigues and Neumann, 2003):

$$-\frac{d[M]}{dt} = R_P = k_P[M][P\bullet] = k_P[M]_0(1-\alpha)[P\bullet] \quad (3)$$

The total concentration of polymeric radicals [P•] is given by:

$$\frac{d[P\bullet]}{dt} = R_i - R_t = \phi_i I_a - k_t[P\bullet]^b \qquad (4)$$

where $\varnothing_i$ is the quantum efficiency of the initiation and I_a is the light intensity. The exponent b is equal to 2 for bimolecular termination processes, and equal to 1 either for unimolecular termination or radical trapping mechanisms.

Phenomenological models

Phenomenological models were developed assuming that only one reaction can represent the whole curing process, and are given by the following equation:

$$\frac{d\alpha}{dt} = k_c(T) f(\alpha) \qquad (5)$$

where dα/dt is the reaction rate, f(α) is a function of conversion, and k_c(T), the chemical-controlled rate constant, is a function of temperature.

The simplest and most common analytical form of f(α) is given by:

$$f(\alpha) = (1-\alpha)^n \qquad (6)$$

where n is a constant that corresponds to the reaction order. Substituting equation (6) into equation (5), the following equation is obtained:

$$\frac{d\alpha}{dt} = k_c(T)\cdot(1-\alpha)^n \qquad (7)$$

which corresponds to the "so-called" *n*th order kinetic model.

The rate constant is supposed to observe an Arrhenius law, so it can be expressed by:

$$k_c(T) = k_0 \exp\left(\frac{-E}{R(\Delta + T)}\right) \qquad (8)$$

where k_0 is a pre-exponential factor or frequency factor, E is the activation energy, R is the gas constant, T is the temperature in ºC T+Δ is the absolute temperature (Δ=273ºC), indicated from now on by T_{abs}.

Considering the rate constant expression, equation (7) is given by:

$$\frac{d\alpha}{dt} = k_0 \exp\left(-\frac{E}{R\ T_{abs}}\right)\cdot(1-\alpha)^n \qquad (9)$$

For an isothermal reaction, the *n*th order kinetic by expression given by equation (9) predicts a maximum of the reaction rate, at time t = 0. However, if an isothermal process is characterised by a thermogram showing a maximum value of the reaction rate at any point, rather than the reaction starting point, the *n*th order kinetic model cannot be applied. In these cases, the *n*th order models are usually replaced by the so-called autocatalytic models, which were initially proposed by Kamal and Sourour (1972, 1973):

$$\frac{d\alpha}{dt} = \left(k_{c1}(T) + k_{c2}(T)\ \alpha^m\right)\ (1-\alpha)^n \qquad (10)$$

where k_{c1}(T) and k_{c2}(T) are rate constants expressed by equation (8), m and n are constants whose sum is the overall reaction order. The order of the reaction, according to its definition, indicates the number of atoms, molecules or reactive groups whose concentration determines the reaction rate.

The Kamal model was modified by Bártolo (2001) to incorporate the effects of radiation over the curing process. In Bártolo's model, the kinetics parameters are defined as functions of temperature, resin composition, light intensity

and fractional conversion. Diffusion-controlled effects characterising the vitrification phenomena are also considered. The models developed by Bártolo (2001), are appropriate to simulate conventional stereolithography, infrared stereolithography and stereo-thermo-lithography, are given by (Bártolo 2001, 2006, and 2007):

- thermal-initiated curing reactions:

$$\frac{d\alpha}{dt} = \frac{1}{1+\exp[\ \xi\ (\alpha - \alpha_d)]}\ \varphi\ \ \exp\left(\frac{-E}{R\ T_{abs}}\right)\ [\beta]^q\ \alpha^m\ (1-\alpha)^n \qquad (11)$$

- photo-initiated curing reactions:

$$\frac{d\alpha}{dt} = \frac{1}{1+\exp[\ \xi\ (\alpha-\alpha_d)]}\ \varphi\ I^p\ \ \exp\left(\frac{-E}{R\ T_{abs}}\right)\ [\beta]^q\ \alpha^m\ (1-\alpha)^n \qquad (12)$$

where ξ is the diffusion constant, α_d is the critical value of fractional conversion corresponding to the beginning of vitrification, φ is a pre-exponential factor, I is the light intensity, E is the activation energy, R is the gas constant, T_{abs} is the absolute temperature, p and q are constants, and m and n exponents represent the reaction order.

These models were experimentally tested with unsaturated polyester resins (Bártolo, 2001), and recently extended to other radical and cationic systems, as well to highly reinforced resins (Gaspar, 2006).

2.2 *Stereolithographic simulation*

The prediction of the global temperature, due to the chemical reaction and the laser irradiation effect, is crucial for the design optimisation of any sterelithographic system. The value of the fractional conversion, obtained at a specific time during the course of the chemical reaction represents another important parameter to be determined.

To simulate the curing process of stereolithographic applications a proper model was developed based on the fundamental physical and chemical phenomena that govern the behaviour of a thermosetting material in both thermal-initiated and photo-initiated curing applications. A number of assumptions were made for its development:

- a simplified isotropic/orthotropic material, whose properties are independent of the cure stage;
- a well mixed and homogeneous initial system;
- any optical scattering effects and the flow of material due to convection or diffusion are considered negligible;
- the internal heat generation is due to the heat of polymerisation;
- the absorption of radiation is defined by the Beer-Lambert law.

According to the proposed model, the temperature field of the polymeric material can be described by the following heating conduction equation:

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}+\frac{\dot{q}}{k}\right)=\rho\, c\,\frac{\partial T}{\partial t} \qquad (13)$$

where k is the thermal conductivity, T is the temperature, ρ is the density, c is the specific heat of the material and t is the time.

The solution of equation (13) requires the knowledge of both the initial temperature, T_i, and the initial value of the fractional conversion, α_i. The solution of equation (13) must satisfy the following boundary conditions:

(1) Specified temperature:

$$T_s = T(x,y,z,t) \qquad (14)$$

(2) specified light intensity:

$$k_n\frac{\partial\ T}{\partial\ n} - I(x,y,z,t) = 0 \qquad (15)$$

(3) convection boundary conditions:

$$k_n\frac{\partial\ T}{\partial\ n} + h\left(T(x,y,z,t) - T_\infty\right)\ =\ 0 \qquad (16)$$

where T_s is the specified temperature, I is the light intensity, $\partial T/\partial n$ is the derivative of temperature in the normal direction to the surface, h is the coefficient of heat transfer, and T_∞ is the temperature of the surrounding space.

The light intensity was assumed to have a gaussian intensity distribution:

$$I_s\ (x,y,z,t)\ =\ I_0\ \ \exp\left[-2\left(\frac{s(x,y,z,t)}{w_0}\right)^2\right] \qquad (17)$$

where I_0 is the peak light intensity, s(x,y,z,t), representing the position in time of a point under irradiation, and w_0 the beam radius. The decrease in light intensity with depth is assumed to obey to the Beer-Lambert law according to the following equation:

$$I_d\ (x,y,z,t)\ =\ I_s\ (x,y,z,t)\ \exp\left[-\varepsilon\ [\beta]\ y\right] \qquad (18)$$

where I_d represents the variation of light intensity along the thickness of the resin layer, ε is the absorptivity of the initiator, $[\beta]$ is the initiator concentration and y represents the penetration depth.

The kinetic models considered are:

- thermal-initiated curing reactions:

$$\frac{d\alpha}{dt} = \varphi\ \ \exp\left(\frac{-E}{R\ T_{abs}}\right)\ [\beta]^q\ \alpha^m\ (1-\alpha)^n \qquad (19)$$

- photo-initiated curing reactions:

$$\frac{d\alpha}{dt} = \varphi\ I^p\ \ \exp\left(\frac{-E}{R\ T_{abs}}\right)\ [\beta]^q\ \alpha^m\ (1-\alpha)^n \qquad (20)$$

The numerical solution of equation (13), subjected to initial conditions and boundary conditions, from equations (14) to (16), involves two approximation stages:

- a spatial approximation using the Galerkin method
- a temporal approximation using the Crank-Nicolson method.

The discretisation of equation (13) in space is obtained by the discretisation of the generic domain Ω into an

appropriate collection of finite elements, Ωei, connected at nodes as follows:

$$\Omega = \sum_{i=1}^{k} \Omega e_i \tag{21}$$

with k being the number of elements and Ωe representing each element domain.

The interpolation functions N_i are used to interpolate the temperature:

$$T(\upsilon,t) = \sum_{i=1}^{j} N_i(\upsilon)\ T_i(t) \tag{22}$$

where j represents the number of nodes of the elements considered and T_i are the nodal temperatures. After applying the Greens's theorem and some mathematical manipulations, the following equation is obtained:

$$[C]\cdot\{\dot{T}\} + [K]\cdot\{T\} = \{R\} \tag{23}$$

where C is the heat capacity matrix, K is the conductivity matrix and R is the equivalent nodal heat flow vector. Matrix elements in equation (23) are, for the bi-dimensional case considered in this paper, given by:

$$C_{ij} = \int_{\Omega_e} N_i \rho C N_j dxdy \tag{24}$$

$$K_{ij} = \int_{\Omega_e} \left[a(x,y)\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + b(x,y)\cdot \right.$$
$$\left. \cdot\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + c(x,y) N_i N_j \right] dxdy \tag{25}$$

$$R_i = \int_{\Omega_e} N_i \rho H \frac{d\alpha}{dt} dxdy + \int_{\Gamma_e} N_i I d\Gamma +$$
$$+ \int_{\Gamma_e} N_i h T_\infty d\Gamma \tag{26}$$

Considering the temporal approximation, the following equation is obtained:

$$\left[\hat{K}\right]_{S+1}\{u\}_{S+1} = \left\{\hat{F}\right\}_{S,S+1} \tag{27}$$

where:

$$\left[\hat{K}\right]_{S+1} = [M] + \theta\Delta t[K]_{S+1}, \text{ and}$$
$$\left\{\hat{F}\right\} = \Delta t\left(\theta\{F\}_{S+1} + (1-\theta)\{F\}_S\right) +$$
$$+\left([M] - (1-\theta)\Delta t[K]_S\right)\{u\}_S$$

After the assembly and the imposition of boundary conditions, equation (27) is solved at each time step, for the nodal values. The resolution starts at t=0 with the definition of initial temperature values. Depending on θ values of different methods can be considered:

$$\theta = \begin{cases} 0 & \text{Progressive differences method} \\ 1/2 & \text{Crank-Nicolson method} \\ 2/3 & \text{Galerkin method} \\ 1 & \text{Regressive differences method} \end{cases}$$

Linear rectangular elements were considered for this study and the concept of isoparametric formulation used. Fractional conversions are evaluated by integrating the kinetic models using the fourth order, fixed step-size Runge-Kutta technique with respect to time.

3. Results

3.1 *Simulation of isothermal reactions in thermo-initiated processes*

The proposed model was used to predict curing profiles for samples of unsaturated polyester resin (UP resin) containing 2 and 0.5 wt% of thermo-initiator, cured at different temperatures. The effect of the curing temperature and initiator concentration on the curing kinetics is illustrated by Figures 1 and 2. Results show that, after an initial period, called induction time (Fouassier, 1995), more important for samples cured at low temperatures and containing low thermo-initiator quantities, the conversion rate increases rapidly followed by a progressive slowing down, until the cure profile reaches a plateau corresponding to the maximum value of the fractional conversion. This progressive slowing down is due to diffusion constraints on the mobility of the reacting species. The viscosity increases as the solidification progresses and, at the vitrification point, the viscosity of the medium is so high that the diffusivity of reactive groups becomes seriously restricted, so the reaction eventually stops. Results also show that, by increasing both temperature and initiator concentration both the reaction rate and the fractional conversion increase while the induction time decreases as well the need for pos-curing operations.

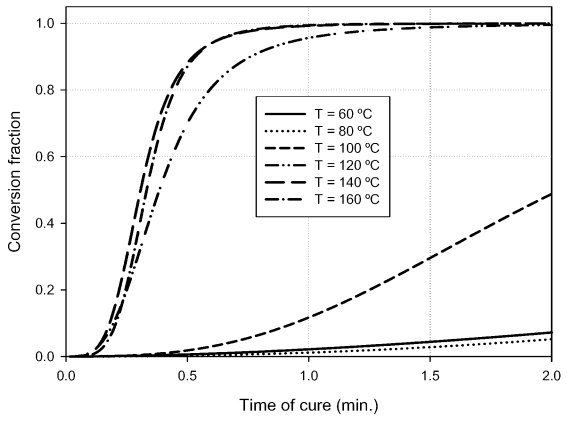


Fig. 1. Profiles of fractional conversion versus curing time for UP resin samples containing 2wt% of thermal-initiator cured at different temperatures.

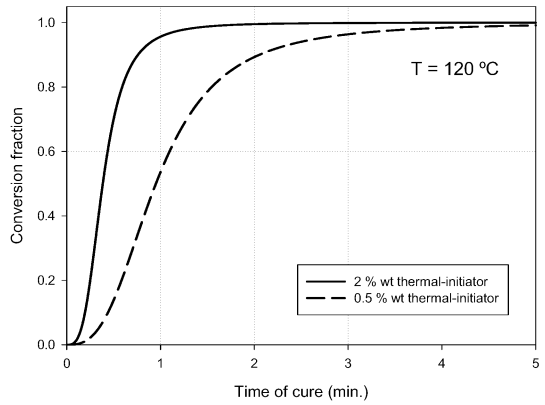


Fig. 2. Profiles of fractional conversion versus curing time for UP resin samples containing 2wt% and 0,5wt% of thermal-initiator cured at 120º C.

The variation of the reaction or conversion rate, as a function of time for different curing temperatures is illustrated by Figure 3. It is possible to observe that, at the lowest temperatures, the conversion rate is low but increases rapidly by increasing the curing temperature. Figure 4 shows the variation of conversion rate with the fractional conversion for different curing temperatures. Results show that, by increasing the curing temperature, the maximum value of conversion rate occurs also for higher fractional conversion values, owing to an increase on the mobility of reactive species and reduction of vitrification effects.

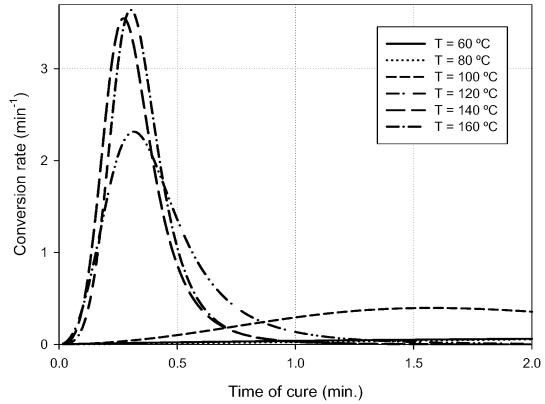


Fig. 3. Conversion rate versus curing time for UP resin samples containing 0,5wt% of thermal-initiator cured at different curing temperatures.

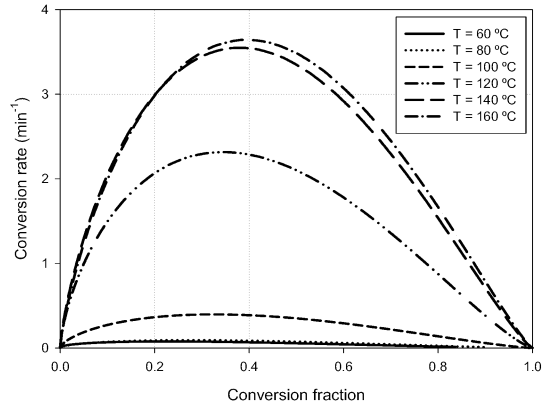


Fig. 4. Conversion rate versus fractional conversion for UP resin samples containing 0,5wt% of thermal-initiator cured at different curing temperatures.

3.2 *Simulation of photo-initiated processes in polymeric systems submitted to constant intensities of radiation*

In this case UP resin samples containing 0.2 wt% of photo-initiator were irradiated at different light intensities. All curing reactions are conducted at 25 ºC. The obtained conversion profiles are shown in Figure 5. Results show that, after an induction time more evident for low light intensity values, the reaction rate increases rapidly. The maximum conversion is earlier achieved for high light intensities.

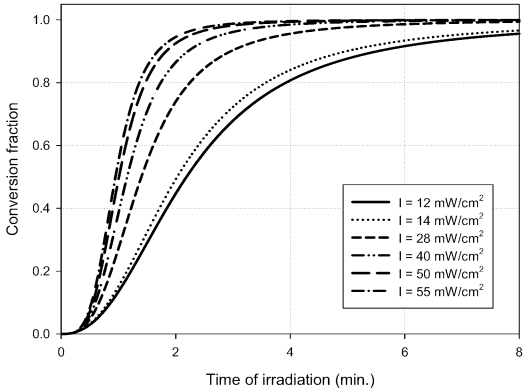


Fig. 5. Fractional conversion versus curing time for UP resin samples containing 0,2 wt% of photo-initiator cured at different light intensities.

3.3 *Simulation of photo-initiated processes at polymeric systems submitted to a radiation profile*

The model was used to predict the necessary time to achieve a complete cure for a layer UP resin containing 0.2 wt% of photo-initiator, which was irradiated at room temperature using an UV lamp. A resin layer with 0.9 cm thickness and 0,7 cm length was considered. In addition, the centre of the layer and the centre of the light beam, which has a Gaussian distribution profile (see Figure 6), were assumed to be coincident. Consequently, only one half of the resin layer was considered for simulation purposes due to the geometry and light distribution symmetry of the problem. Figure 7 shows contour plots of fractional conversion and temperature for different exposure times. The fractional conversion contour indicates a conical variation, which can be an important factor in terms of the quality of the final models. Figures 8 and 9 illustrate this effect on the curing process of increasing light intensity and photo-initiator concentration. The reaction rate increases by increasing both light intensity and initiator concentration as expected. However, the increase of the photo-initiator concentration decreases the light penetration depth.

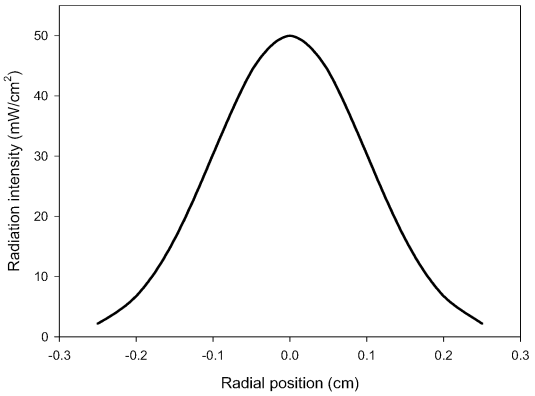
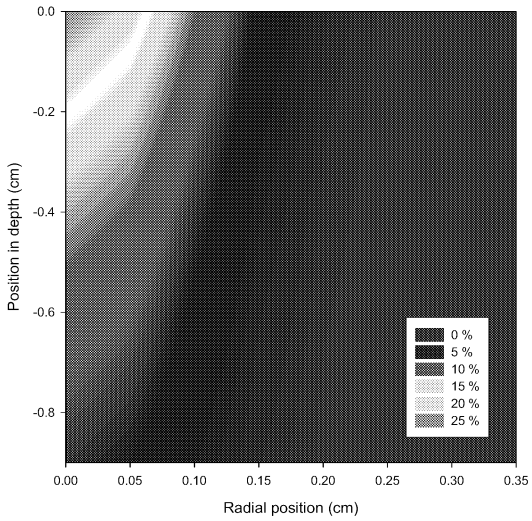


Fig. 6. Light intensity profile at the resins' surface.



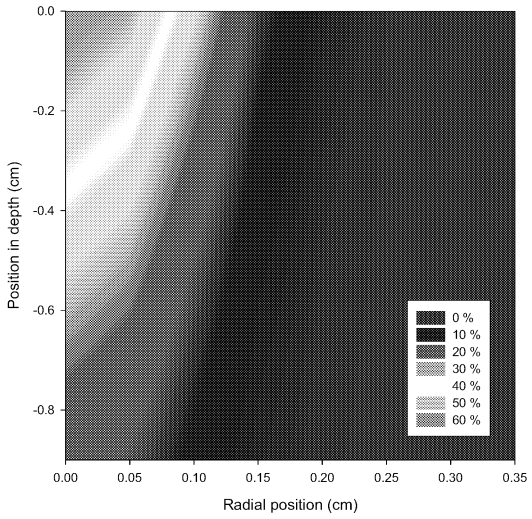


Fig. 7. Contour plot showing the variation of fractional conversion as a function of position. Irradiation times: (a) 10 and (b) 15 min. Light intensity: 50 mW/cm^2 of intensity. UP sample containing 0.2 wt% of photo-initiator.

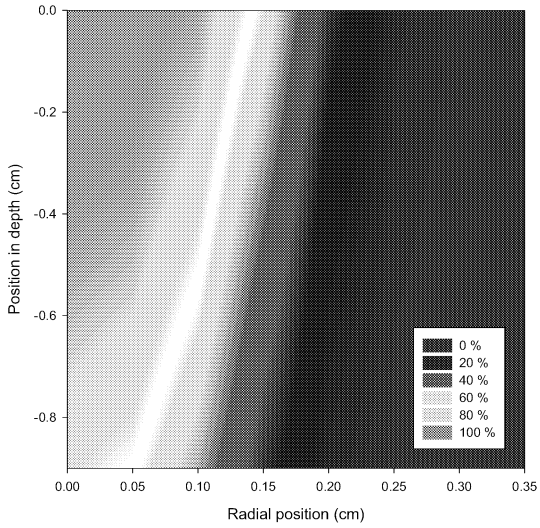
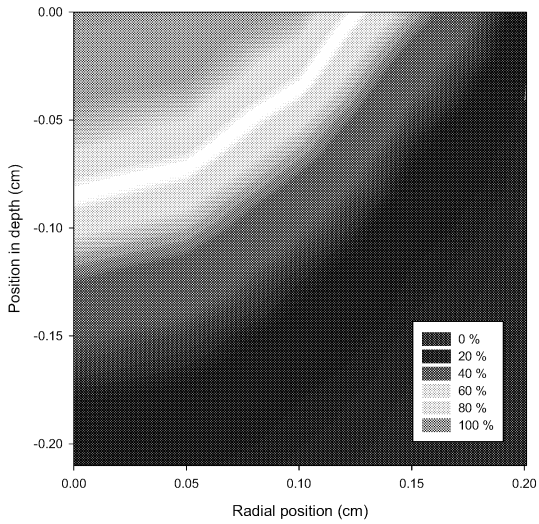


Fig. 8. Contour plot showing the variation of fractional conversion as a function of position. Irradiation time: 10 min. Light intensity: 100 mW/cm^2 of intensity. UP sample containing 0.2 wt% of photo-initiator.



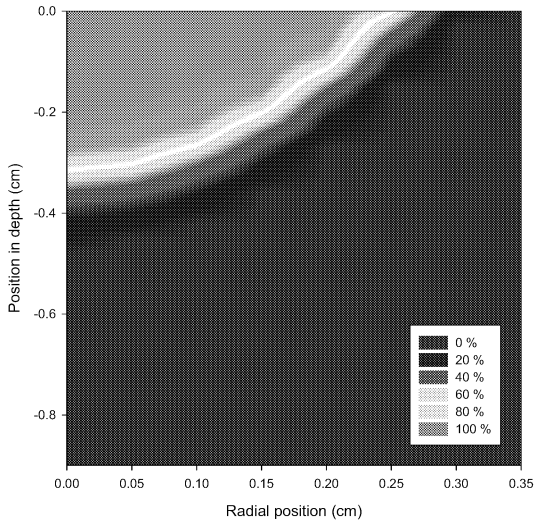


Fig. 9. Contour plot showing the variation of fractional conversion as a function of position. Irradiation times: (a) 1 and (b) 5 min. Light intensity: 100 mW/cm^2 of intensity. UP sample containing 2 wt% of photo-initiator.

4. Conclusions

A simulation model using the finite element method was proposed, based on fundamental physical and chemical

phenomena governing the behaviour of a thermosetting material for both thermal-initiated and photo-initiated curing applications. This model describes the major events occurring during the chemical process of liquid resin solidification, enabling to control and optimise the most important curing parameters. Results show that, by increasing the temperature, light intensity and initiator, the reaction rate increases while the induction time decreases. For radiation profiles, the conversion contour shows a conical shape, which is relevant for part quality. The penetration depth decreases with the increase of photo-initiator according to the Beer-Lambert law.

References

Bártolo, P.J. (2001): *Optical Approaches to Macroscopic and Microscopic Engineering,* PhD thesis, University of Reading, UK

Bártolo, P.J. and Jardini, A.L. (2004): Stereolithographic Processes: Materials, Techniques and Applications. In: *Proceedings of 10th European Forum on Rapid Prototyping*. Paris, France.

Bártolo, P.J. (2006): Computer Simulation of Stereolithographic Curing Reactions: Phenomenological versus Mechanistic Approaches. *Annals of CIRP*, 55, No. 1, pp. 221-226

Bártolo, P.J. (2007): Photo-curing Modelling: Direct Irradiation. *International Journal of Advanced Manufacturing Technology*, 32, pp. 480-491

Fouassier, J.-P. (1995): *Photoinitiation, Photopolymerization, and Photocuring – Fundamentals and Applications*. Hanser

Gaspar, J. (2006): *Study of the Flow and Curing Behaviour of the Reinforced Polymeric Systems for Stereolithography* [master thesis, in Portugal]. School of Engineering, Minho University, Guimarães

Odian, G. (1981): *Principles of Polymerization*. John Wiley & Son, New York (2nd ed.)

Painter, P.C. and Coleman, M.M. (1997): *Fundamentals of Polymer Science – An Introductory Text*. Technomic Publishing (2nd ed.)

Rodrigues, M.R. and Neumann, M.G. (2003): Fotopolimerização: Príncipios e Métodos. *Polímeros: Ciência e Tecnologia*, 13, No. 4, pp. 276-286

Salonitis, K., Tsoukantas, G. and Stavropoulos, A., (2003): A Critical Review os Stereolithography Process Modeling. In: P. J. Bártolo *et al* (editors): *Advanced Research in Virtual and Rapid Prototyping*. pp. 377-384

Souror, S. and Kamal, M.R. (1972). *SPE ANTEC Technical Paper*, 18, No. 93

Souror, S. and Kamal, M.R. (1973): Kinetics and Thermal Characterization of Thermoset Cure. *Polymer Engineering & Science*, 13, No. 1, pp. 59-64

Yamaguchi, K. and Nakamoto, T. (1998): Microfabrication by UV Laser Photopolymerization. *Memoirs of the School of Engineering, Nagoya University*, 50, 1998, No 1/2, pp. 33-82

Yousefi, A., Lafleur, P.G. and Gauvin, R. (1997): Kinetic Studies of Thermoset Cure Reactions: A Review. *Polymer Composites*, 18, No 2, pp. 157-168

COMPLEX SYSTEMS SIMULATION

The aim of this workshop/track is to concern itself with the use of emergent computing and self-organization modelling within various applications of complex systems. We focus our attention both on the innovative concepts and implementations to model self-organizations, but also the relevant applicative domains, which can use them in an efficient way

The first part deals with various conceptual or practical tools allowing the relevant description of self-organization processes. The second part deals with specific applications on transportation and traffic flow. The third part completes the applicative issues by various topics on mobile robotic, urban systems, cognitive modelling and classification.

Cyrille Bertelle
Track Organizer

LITIS (EA 4108)
UFR Sciences et Techniques
25 rue Ph. Lebon - BP 540
76058 Le Havre Cedex - FRANCE

CONCEPTUAL MODELS AND TOOLS FOR COMPLEXITY ANALYSIS

A simulation-optimisation framework for determining a best strategy in inventory decision making and demand forecasting for intermittent demand

Katrien Ramaekers and Gerrit K. Janssens
Transportation Research Institute
Hasselt University
Wetenschapspark 5 bus 6, 3590 Diepenbeek, Belgium
e-mail: {katrien.ramaekers;gerrit.janssens}@uhasselt.be

Keywords: intermittent demand, forecasting, inventory management, simulation, optimisation, supply uncertainty

Abstract

Demand forecasting is one of the most crucial aspects of inventory management. For intermittent demand, i.e. demand peaks follow several periods of zero or low demands, forecasting is difficult. Furthermore, uncertainty in supply is often not taken into account. A simulation model is used to study a single-product inventory system facing demand of the intermittent type and dealing with an unreliable supplier. In this paper, a research approach is described to find a best strategy in combining inventory decision making and demand forecasting for intermittent demand.

1 Introduction

Inventory systems have to cope with uncertainty in demand. The inventory control literature mostly makes use of the Normal or Gamma distribution for describing the demand in the lead-time. The Poisson distribution has been found to provide a reasonable fit when demand is very low (only a few pieces per year). Less attention has been paid to irregular demand. This type of demand is characterised by a high level of variability, but may be also of the intermittent type, i.e. demand peaks follow several periods of zero or low demands. In practice, items with intermittent demand include service or spare parts and high-priced capital goods. A common example of such goods are spare parts for airline fleets.

Demand forecasting is one of the most crucial aspects of inventory management [8]. However, for intermittent demand, forecasting is difficult, and errors in prediction may be costly in terms of obsolescent stock or unmet demand [7]. The standard forecasting method for intermittent demand items is considered to be Croston's method [2]. This method builds estimates taking into account both demand size and the interval between demand occurrences. Despite the theoretical superiority of such an estimation procedure, empirical evidence suggests modest gains in performance when compared with simpler forecasting techniques [6]. Furthermore, the choice of the forecasting method can have an impact on the inventory management policy that is best used.

Studies in literature mainly focus on uncertainty in demand. In many inventory models the continuous availability of supply at any time in the future is an implicit assumption. However, uncertainty is also present at the supply side. This type of uncertainty occurs in delivery time, in interruption of delivery during a certain period, or in mismatches in order and delivery in terms of quality or quantity. In this paper, the focus is on uncertainty in availability. The supplier alternates randomly between an available and an

unavailable state. When the supplier is available, the order is delivered after the usual lead time. When the supplier is unavailable, the order is executed when the supplier turns available again.

Because of the uncertainty present in the inventory system, often mathematical models cannot accurately describe the system. Therefore, simulation models will be used. In preliminary research a simulation model is built and optimised to obtain the best strategy in combining inventory decision making and demand forecasting for intermittent demand when there is no uncertainty in supply. In this paper, the same simulation model is used but the assumption of no uncertainty in supply is relaxed. First, the results of the optimal policies found for a reliable supplier are compared to results when applying the same policies for an unreliable supplier. Next, a new best strategy in combining inventory decision making and demand forecasting is determined for the situation of an unreliable supplier.

2 Simulation model and research approach

2.1 Simulation model

The study focuses on a single-product inventory system facing demand of the intermittent type. The simulation model is developed in Microsoft Excel spreadsheets and uses VBA. The simulation model starts by generating intermittent demand as described in the previous section. Next, the inventory system is simulated for 52 periods. At each review-time, a demand forecast and an order decision are made. The total costs of the inventory system are determined. 10 replications are made for each simulation run.

To generate intermittent demand, demand occurrence and demand size are separately generated. The demand occurrence is generated according to a first-order Markov process with transition matrix

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix},$$

where p_{00} is the probability of no order in the next period when there has been no order in this period and p_{10} is the probability of no order in the next period when there has been an order in the current period. Individual order sizes are generated using a Gamma distribution with shape parameter γ and scale parameter β.

The standard forecasting method for intermittent demand items is considered to be Croston's method. However, in practice, single exponential smoothing and simple moving averages are often used to deal with intermittent demand. These three forecasting methods are compared.

In this research, two periodic review models are used. The first one is the (R, s, S) system. This means that every R units of time, the inventory level is checked. If it is at or below the reorder point s, a sufficient quantity is ordered to raise it to S. The second system (R, s, Q) is similar to the (R, s, S) system but uses a fixed order quantity Q instead of an order-up-to-level S.

A deterministic lead-time L is assumed. The following costs are considered: unit holding cost per period C_h, ordering cost C_o and unit shortage cost per period C_s. The simulation starts with an initial inventory level I_0.

2.2 Experimental design

The parameters of the inventory system to optimise include both qualitative and quantitative factors. The experimental design includes two qualitative factors: the forecasting method and the inventory management policy. In addition, depending on the choice of the qualitative factors, a set of quantitative factors are part of the experimental design. If the (R, s, Q) inventory management policy is used, the safety stock SS and order quantity Q are the parameters to optimise. If the (R, s, S) inventory management policy is used, the safety stock SS and order-up-to-level S are the optimising parameters. For single exponential smoothing and Croston's method, the smoothing parameter α is optimised and for moving averages, the weights of the past values are optimised.

2.3 Research approach

Because of the dependence of the quantitative factors on the choice of the qualitative factors, we use for the optimisation the research approach described in this section.

For every combination of forecasting method, inventory management policy and review period, the optimal values of the quantitative factors are determined. The total costs of the inventory system are optimised using tabu search. Tabu search is shortly described below. Once the optimal values are found, the best combination of forecasting method, inventory management policy and review period is chosen.

Tabu search uses a local or neighbourhood search procedure to iteratively move from one solution to the next in the neighbourhood of the first, until some stopping criterion has been satisfied. To explore regions in the search space that would be left unexplored by the local search procedure and escape local optimality, tabu search modifies the neighbourhood structure of each solution as the search progresses. The solutions admitted to the new neighbourhood are determined through the use of special memory structures. Tabu search uses both long-term and short-term memory, and each type of memory has its own special strategies [3, 4].

Tabu search is a heuristic optimisation technique developed specifically for combinatorial problems. Very few works deal with the application to the global minimization of functions depending on continuous variables. The method we propose in this paper is based on [1, 5]. The purpose in these papers is to keep as close as possible to original tabu search. Two issues must be examined: the generation of current solution neighbours and the elaboration of the tabu list.

To define a neighbourhood of the current solution, a set of hyperrectangles is used for the partition of the current solution neighbourhood. The k neighbours of the current solution are obtained by selecting one point at random inside each hyperrectangular zone.

Once a new current solution is determined, the immediate neighbourhood of the previous solution is added to the tabu list.

3 Experimental environment

The experimental environment contains the uncontrollable factors of the inventory system: the costs of the inventory system and the parameters for generating intermittent demand. These factors can have an effect on the results that are obtained. The research approach described above, is executed using a single combination of the costs of the inventory system and demand. A fractional factorial design of 16 experimental points is set up for these factors and the optimisation phase is repeated for each experimental point.

Demand occurrence is generated using a first-order Markov process with transition matrices:

$$\mathbf{P_1} = \begin{pmatrix} 0.7875 & 0.2125 \\ 0.85 & 0.15 \end{pmatrix}$$

or

$$\mathbf{P_2} = \begin{pmatrix} 0.5667 & 0.4333 \\ 0.65 & 0.35 \end{pmatrix}.$$

They correspond with a probability of 20% to have demand in a certain period for the first matrix and a probability of 40% to have demand in a period for the second matrix. The *size of demand* is generated using a Gamma distribution with 4 different combinations of the scale parameter γ and the shape parameter β. These values are summarized in Table 1.

Level	γ	β
1	6	1
2	12	1
3	3	2
4	24	0.5

Table 1: Parameters of the Gamma distribution

The levels of the *costs* of the inventory system are given in Table 2. The initial inventory level I_0 equals 5.

Level	C_o	C_h	C_s
1	100	2	5
2	200	4	10

Table 2: Levels for the costs of the inventory system

Exp	C_o	C_h	C_s	Freq	γ	β
1	200	4	10	0.4	12	1
2	100	4	5	0.4	12	1
3	200	2	5	0.4	24	0.5
4	100	2	10	0.4	24	0.5
5	200	2	5	0.4	3	2
6	100	2	10	0.4	3	2
7	200	4	10	0.4	6	1
8	100	4	5	0.4	6	1
9	200	2	10	0.2	12	1
10	100	2	5	0.2	12	1
11	200	4	5	0.2	24	0.5
12	100	4	10	0.2	24	0.5
13	200	4	5	0.2	3	2
14	100	4	10	0.2	3	2
15	200	2	10	0.2	6	1
16	100	2	5	0.2	6	1

Table 3: Experimental design for uncontrollable factors

The fractional factorial design is shown in Table 3. This fractional factorial design makes it possible to determine the impact of uncontrollable factors as the cost structure and the demand.

4 Results

Firstly, results are compared for a reliable and an unreliable supplier. For 16 combinations of uncontrollable factors, the best strategy in combining inventory decision making and demand forecasting for a reliable supplier is determined in preliminary research (Table 4). Depending on the experimental environment, two options for optimal strategies can be distinguished: an order-up-to level inventory management policy with an order-up-to level equal to 1 and a reorder point equal to 0 or an inventory management policy with a fixed order quantity $Q > 1$ or an order-up-to level $S > 1$ and a reorder point equal to 0. This optimal strategy will also be used to determine output measures for the inventory system with an unreliable supplier.

Exp	Best strategy
1	MA/FOQ; ROP=0; Q=25
2	ES/OUL; ROP=0; S=1
3	MA/OUL; ROP=0; S=30
4	MA/OUL; ROP=0; S=25
5	MA/FOQ; ROP=0; Q=20
6	MA/OUL; ROP=0; S=15
7	MA/FOQ; ROP=0; Q=15
8	ES/OUL; ROP=0; S=1
9	MA/FOQ; ROP=0; Q=20
10	ES/OUL; ROP=0; S=1
11	CR/OUL; ROP=0; S=1
12	ES/OUL; ROP=0; S=1
13	MA/OUL; ROP=0; S=1
14	MA/OUL; ROP=0; S=1
15	MA/OUL; ROP=0; S=15
16	ES/OUL; ROP=0; S=1

Table 4: Optimal results based on Tabu search for a reliable supplier

Tables 5 contains 95% confidence intervals (CI) for the difference in total costs between a reliable and an unreliable supplier.

For eight of the experimental points, the difference in costs between the reliable and the unreliable case is significant. The alternative with the unreliable supplier has higher total costs than the one with the reliable supplier. The experimental points with a significant difference in costs are those experimental points for which the best strategy for a reliable supplier has an order-up-to-level of 1. For the other experimental points, the best strategy for a reliable supplier has an order-up-to-level or fixed order quantity of 15 or more.

A new optimal combination of forecasting method and inventory management policy is determined for the inventory management system with intermittent demand and an unreliable supplier. Furthermore, the optimal settings for the safety stock and the fixed order quantity or

Exp	CI
1	-150.68;149.88
2	-516.98;-248.38
3	-74.65;97.89
4	-135.08;-5.64
5	-25.81;144.886
6	-65.95;74.03
7	-103.07;139.35
8	-369.92;-144.44
9	-117.4;131.08
10	-359.4;-124.36
11	-679.87;-279.44
12	-597.74;-290.24
13	-459.96;-108.36
14	-507.49;-263.19
15	-100.16;91.3
16	-232.57;-33.19

Table 5: Confidence intervals for comparing costs of a reliable and an unreliable supplier

Exp	Best strategy
2	MA/FOQ; ROP=0; Q=18
8	MA/FOQ; ROP=0; Q=12
10	ES/OUL; ROP=0; S=1
11	CR/OUL; ROP=0; S=1
12	MA/OUL; ROP=0; S=1
13	MA/FOQ; ROP=0; Q=10
14	ES/OUL; ROP=0; S=1
16	MA/FOQ; ROP=0; Q=10

Table 6: Results of Tabu search for an unreliable supplier

order-up-to-level are determined.

Table 6 shows the best strategy for the inventory system with an unreliable supplier.

The best strategy for the inventory system with intermittent demand and no uncertainty in supply is an order-up-to-level inventory management policy with $S = 1$. When the results in Table 6 are compared to the results in Table 4, the experimental points can be divided in two categories. For the experimental points 2, 8, 13 and 16, the best strategy is an inventory management policy with a fixed order quantity Q equal to 10 or more. When a fixed order quantity inventory management policy is used, moving averages is always the best forecasting method. For the experimental points 10, 11, 12 and 14, the best strategy is an order-up-to-level inventory management policy with S=1, which is the same strategy as found for the inventory system with a reliable supplier. When the order-up-to-level equal to 1 is the best inventory management policy, no preference for a specific forecasting method can be found.

5 Conclusions and further research

In this paper, the simulation model developed in preliminary research is extended to include uncertainty in the supply side. Uncertainty in supply consists of many aspects, the focus of this paper is on uncertainty in availability. The supplier alternates randomly between an available and an unavailable state. When the supplier is available, the order is delivered after the usual lead time. When the supplier is unavailable, the order is executed when the supplier turns available again.

When the best strategy for the inventory system with a reliable supplier is used for the system with an unreliable supplier, 8 of 16 experimental points show a significant difference in total costs. For these 8 experimental points, a new optimal strategy in combining demand forecasting and inventory management decision making is determined using Tabu Search. For four of these points, no better strategy could be determined. For the other four experimental points, it is better to use a fixed order quantity Q of 10 or more instead of an order-up-to-level S equal to 1.

A best strategy in combining inventory decision making and demand forecasting for intermittent demand is determined using the objective of minimising total costs. However, there are situations in which total costs are not the best decision variable. In future work, it can be interesting to determine a best strategy using a performance measure as the output measure to optimise. Multi-objective optimisation, including non-cost but service-oriented performance characteristics, can also be used to

better incorporate the effect on performance measures in the optimal solution.

References

[1] Chelouah R. and Siarry P.: Tabu search applied to global optimization, *European Journal of Operational Research* 123, 2000, pp. 256-270.

[2] Croston J.D.: Forecasting and stock control for intermittent demands, *Operational Research Quarterly* 23, 1972, pp. 289-303.

[3] Dengiz B. and Alabas C.: Simulation optimisation using tabu search, *In Proceedings of the 2000 Winter Simulation Conference*, Orlando, 10-13 December 2000, pp. 805-810.

[4] Glover F.: Tabu search- Part I, *ORSA Journal on Computing* 1(3), 1989, pp. 190-206.

[5] Siarry P. and Berthiau G.: Fitting of tabu search to optimize functions of continuous variables, *International Journal of Numerical Methods in Engineering* 40, 1997, pp. 2449-2457.

[6] Syntetos A.A. and Boylan J.E.: On the bias of intermittent demand estimates, *International Journal of Production Economics* 71, 2001, pp. 457-466.

[7] Syntetos A.A. and Boylan J.E.: The accuracy of intermittent demand estimates, *International Journal of Forecasting* 21, 2005, pp. 303-314.

[8] Willemain T.R., Smart C.N. and Schwarz H.F.: A new approach to forecasting intermittent demand for service part inventories, *International Journal of Forecasting* 20, 2004, pp. 375-387.

Biography

Katrien Ramaekers graduated as Master of Business Economics, option Technology at the Limburg University Centre in 2002. In October 2002, she started as a Ph.D.-student at Hasselt University. Her main research interest is on the integration of simulation and optimisation, especially as a support for complex logistics decision-making and for decision support with limited information in supply and demand. She is a member of the Transportation Research Institute of Hasselt University.

Gerrit K. Janssens received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University within the Faculty of Business Administration. He also holds the CPIM certificate of the American Production and Inventory Control Society (APICS). During the last fifteen years he has repeatedly been visiting faculty member of universities in Thailand, Vietnam, Cambodia and Zimbabwe. His main research interests include the development and application of operations research models in production and distribution logistics.

INVARIANT ALGEBRAIC MANIFOLDS OF DIFFERENTIAL SYSTEMS

Jean-Marc Ginoux and Bruno Rossetto
P.R.O.T.E.E. Laboratory
I.U.T. de Toulon, Université du Sud
B.P. 20132, 83957, La Garde cedex,
France
E-mail: ginoux@univ-tln.fr, rossetto@univ-tln.fr

KEYWORDS

Invariant curves, invariant surfaces, curvature, torsion, Lagutinskii.

ABSTRACT

At the end of the XIXth century various methods (Darboux and Lagutinskii) for determining invariant manifolds of differential systems, i.e., dynamical systems have been developed. These manifolds have been used in order to build the general integral of such systems. Their existence plays a very important role in the stability of dynamical systems. So, their determination has become a problem of great interest. The aim of this work is to present a method based on the use of *Differential Geometry* properties such as curvature and torsion which directly provides the invariant algebraic manifolds equations of degree 1 of such systems. Moreover, identity between this approach and those of Darboux and Lagutinskii is established. Then, various applications to dynamical systems of dimension two and three provide their general integral.

DIFFERENTIAL SYSTEMS

In the following we consider a system of ordinary differential equations defined in a compact E included in $\mathbb{R}$:

$$\frac{d\vec{X}}{dt} = \vec{\Im}\left(\vec{X}\right) \quad (1)$$

with $\vec{X} = \left[x_1, x_2, x_3\right]^t \in E \subset \mathbb{R}^n$

and $\vec{\Im}\left(\vec{X}\right) = \left[f_1\left(\vec{X}\right), f_2\left(\vec{X}\right), f_3\left(\vec{X}\right)\right]^t \in E \subset \mathbb{R}^n$

The vector $\vec{\Im}\left(\vec{X}\right)$ defines a velocity vector field in E whose components f_i which are supposed to be continuous and infinitely differentiable with respect to all x_i and *t*, i.e., are C^∞ functions in E and with values included in $\mathbb{R}$, satisfy the assumptions of the Cauchy-Lipschitz theorem. For more details, see for example (Coddington and Levinson 1955).

A solution of this system is an *integral curve* $\vec{X}\left(t\right)$ tangent to $\vec{\Im}$ whose values define the states of the dynamical system described by the Equation (1). Since none of the components f_i of the velocity vector field depends here explicitly on time, the system is said to be autonomous.

TRAJECTORY CURVES

The integral of the system (1) can be associated with the co-ordinates, i.e., with the position, of a point M at the instant *t*. The total derivative of $\vec{V}\left(t\right)$ namely the instantaneous acceleration vector $\vec{\gamma}\left(t\right)$ may be written, while using the chain rule, as:

$$\vec{\gamma} = \frac{d\vec{V}}{dt} = \frac{d\vec{\Im}}{d\vec{X}}\frac{d\vec{X}}{dt} = J\vec{V} \quad (2)$$

where J is the functional jacobian matrix of system (1).

Osculating Plane

The osculating plane to the *trajectory curve* is the plane passing through a fixed point I and spanned by the instantaneous velocity $\vec{V}$ and acceleration $\vec{\gamma}$ vectors. In the particular case of the dynamical system admits the origin as a fixed point, osculating plane equation may be read:

$$\vec{X}.\left(\vec{V} \wedge \vec{\gamma}\right) = 0 \quad (3)$$

Then, the *integral curve* defined by the vector function $\vec{X}\left(t\right)$ of the scalar variable t representing the trajectory of M can be considered as a *plane* or a *space curve* which has local metrics properties namely *curvature* and *torsion*.

Curvature

The *curvature*, which expresses the rate of changes of the tangent to the *trajectory curve* of system (1), is defined by:

$$\frac{1}{\Re}=\frac{\left\|\vec{\gamma}\wedge\vec{V}\right\|}{\left\|\vec{V}\right\|^3}=\frac{\phi\left(\vec{X}\right)}{\left\|\vec{V}\right\|^3} \quad (4)$$

Torsion

The *torsion*, which expresses the difference between the *trajectory curve* of system (1) and a *plane* curve, is defined by:

$$\frac{1}{\Im}=-\frac{\dot{\vec{\gamma}}\cdot\left(\vec{\gamma}\wedge\vec{V}\right)}{\left\|\vec{\gamma}\wedge\vec{V}\right\|^2}=\frac{\phi\left(\vec{X}\right)}{\left\|\vec{\gamma}\wedge\vec{V}\right\|^2} \quad (5)$$

INVARIANT MANIFOLD

Let ϕ a C^1 function defined in a compact E included in $\mathbb{R}$ and $\vec{X}(t)$ the integral of the dynamical system defined by (1). The Lie derivative is defined as follows:

$$L_{\vec{X}}\phi=\vec{V}\cdot\vec{\nabla}\phi=\sum_{i=1}^{n}\frac{\partial\phi}{\partial x_i}\dot{x}_i=\frac{d\phi}{dt} \quad (6)$$

Theorem:

If $L_{\vec{X}}\phi=0$ then ϕ is first integral of the dynamical system defined by (1). So, ϕ is constant along each trajectory curve and the first integrals are drawn on the level set $\{\phi=\alpha\}$ and where α is a constant.

Proof of this theorem may be found in (Demazure 1989)

■

Darboux Theorem:

An *invariant curve* (*resp. surface*) is defined by $\phi\left(\vec{X}\right)=0$ where ϕ is a C^1 in an open set U and such there exists a C^1 function denoted $k\left(\vec{X}\right)$ and called cofactor which satisfies

$$L_{\vec{X}}\phi\left(\vec{X}\right)=k\left(\vec{X}\right)\phi\left(\vec{X}\right) \quad (7)$$

for all $\vec{X}\in U$

Proof of this theorem may be found in (Darboux 1878)

■

Lagutinskii Theorem:

Let's note $\vec{\chi}=\sum_{i=1}^{n}\frac{\partial}{\partial x_i}\dot{x}_i$ the homogeneous vector field defined by (1), where $\vec{\chi}^k\left(\phi\right)=\vec{\chi}\left(\vec{\chi}^{k-1}\left(\phi\right)\right)$ for any polynomial ϕ.
Then, the equation for the inflection curve, or the first extactic curve which will be denoted by $\varepsilon\left(\vec{\chi}\right)$ is in

Dimension 2:

$$\varepsilon\left(\vec{\chi}\right)=\begin{vmatrix}1 & x & y\\ 0 & \vec{\chi}(x) & \vec{\chi}(y)\\ 0 & \vec{\chi}^2(x) & \vec{\chi}^2(y)\end{vmatrix}=\begin{vmatrix}1 & x & y\\ 0 & \dot{x} & \dot{y}\\ 0 & \ddot{x} & \ddot{y}\end{vmatrix} \quad (8)$$

Dimension 3:

$$\varepsilon\left(\vec{\chi}\right)=\begin{vmatrix}1 & x & y & z\\ 0 & \vec{\chi}(x) & \vec{\chi}(y) & \vec{\chi}(z)\\ 0 & \vec{\chi}^2(x) & \vec{\chi}^2(y) & \vec{\chi}^2(z)\\ 0 & \vec{\chi}^3(x) & \vec{\chi}^3(y) & \vec{\chi}^3(z)\end{vmatrix}=\begin{vmatrix}1 & x & y & z\\ 0 & \dot{x} & \dot{y} & \dot{z}\\ 0 & \ddot{x} & \ddot{y} & \ddot{z}\\ 0 & \dddot{x} & \dddot{y} & \dddot{z}\end{vmatrix} \quad (9)$$

Every algebraic curve of degree 1 invariant by the vector field $\vec{\chi}$ is a factor of $\varepsilon\left(\vec{\chi}\right)$.

Proof of this theorem may be found in (Christopher 2007)

■

According to the previous Darboux and Lagutinskii theorems the following proposition may be established.

Proposition 1: *The location of the points where the local curvature of the trajectory curves integral of a two dimensional dynamical system defined by (1) vanishes provides the equation for the inflection curve, or the first extactic curve denoted $\varepsilon\left(\vec{\chi}\right)$ and given by (8).*

Proof.

While using the Laplace formula for the computation of the determinant (8) we have:

$$\varepsilon\left(\vec{\chi}\right)=\begin{vmatrix}1 & x & y\\ 0 & \dot{x} & \dot{y}\\ 0 & \ddot{x} & \ddot{y}\end{vmatrix}=1\begin{vmatrix}\dot{x} & \dot{y}\\ \ddot{x} & \ddot{y}\end{vmatrix}=\left\|\vec{\gamma}\wedge\vec{V}\right\|$$

■

Proposition 2: *The location of the points where the local torsion of the trajectory curves integral of a three dimensional dynamical system defined by (1) vanishes provides the equation for the inflection curve, or the first extactic curve denoted* $\varepsilon\left(\vec{\chi}\right)$ *and given by (9).*

Proof.

While using the Laplace formula for the computation of the determinant (9) we have:

$$\varepsilon\left(\vec{\chi}\right)=\begin{vmatrix}1 & x & y & z\\ 0 & \dot{x} & \dot{y} & \dot{z}\\ 0 & \ddot{x} & \ddot{y} & \ddot{z}\\ 0 & \dddot{x} & \dddot{y} & \dddot{z}\end{vmatrix}=1\begin{vmatrix}\dot{x} & \dot{y} & \dot{z}\\ \ddot{x} & \ddot{y} & \ddot{z}\\ \dddot{x} & \dddot{y} & \dddot{z}\end{vmatrix}=\dot{\vec{\gamma}}\cdot\left(\vec{\gamma}\wedge\vec{V}\right)$$

■

Note.

In the particular case where the *inflection curve*, or the *first extactic curve* passes through the origin its equation is identical to the one of the osculating plane.

DISCUSSION

The Lagutinskii, Darboux and Differential Geometry methods are thus identical and provide the *first extactic curve* equation of degree 1 (a line in dimension two, a plane in dimension three). The main difference between these methods lies in the methodology. The former are using an inductive approach which consists in supposing that the invariant manifold is a line (resp. a plane) while the latter is using a deductive approach which leads to show that the invariant manifold is a line (resp. a plane). Moreover, in dimension two the identity is obvious since the location of the points where the local curvature vanishes corresponds to the *inflection points locus*. So, such invariant manifolds may then be used to build general integral of differential systems, i.e., nonlinear dynamical systems. The following applications will exemplify the previous propositions.

APPLICATIONS

But, in order to give the most simple and consistent application, let's focus on two classical examples:

- the Volterra-Lotka model
- the Pikovskii-Rabinovich system.

Volterra-Lotka model

The evolution of two species interacting in a predator-prey mode has been described by a pair of nonlinear differential equations proposed independently by Alfred Lotka (1925) and Vito Volterra (1926). If there are various models called Volterra-Lotka it may be noticed that contrary to Lotka which only studied the case of two variables, Volterra considered in his book: *Leçons sur la théorie mathématique de la lutte pour la vie*, a set of *n* variables (species) interacting in all types of modes (predation, competition, cooperation). One of the classical Volterra-Lotka model may be written:

$$\vec{V}\begin{pmatrix}\dfrac{dx}{dt}\\ \dfrac{dy}{dt}\end{pmatrix}=\vec{\Im}\begin{pmatrix}f\left(x,y\right)\\ g\left(x,y\right)\end{pmatrix}=\begin{pmatrix}x\left(a_1+b_{11}x+b_{12}y\right)\\ y\left(a_2+b_{21}x+b_{22}y\right)\end{pmatrix}\quad(10)$$

with a_i and b_{ij} real parameters and where the functions *f* and *g* are infinitely differentiable with respect to all x_i and *t*, i.e., are C^∞ functions in a compact E included in $\mathbb{R}^2$ and with values in $\mathbb{R}$. Let's consider that parameters a_1 and a_2 are null then we have a homogeneous differential system, the location of the points where the local curvature (4) vanishes leads to the following equation:

$$\phi\left(\vec{X}\right)=xy\left(x\left(b_{11}-b_{21}\right)+y\left(b_{12}-b_{22}\right)\right)=0\quad(11)$$

The equation for the inflection curve, or the first extactic curve (8) provides exactly the same result. Moreover, the Lie derivative of ϕ leads to $L_{\vec{X}}\phi=Tr\left[J\right]\phi\left(\vec{X}\right)$, i.e.,

$$L_{\vec{X}}\phi=\left[x\left(2b_{11}+b_{21}\right)+y\left(b_{12}+2b_{22}\right)\right]\phi\left(\vec{X}\right)\quad(12)$$

So, if parameters are such that:

$$\begin{cases}2b_{11}+b_{21}=0\\ b_{12}+2b_{22}=0\end{cases}\quad(13)$$

Then, ϕ is first integral of the Volterra-Lotka model (10).

Pikovskii-Rabinovich system

This system is a three-wave interaction model, where *x*, *y* and *z* are real variables, v_1, v_2 and v_3 are the damping rates, and *k* is proportional to the driving amplitude of the feeder wave; (see Pikovskii 1978)

$$\vec{V}\begin{pmatrix}\dfrac{dx}{dt}\\ \dfrac{dy}{dt}\\ \dfrac{dz}{dt}\end{pmatrix}=\vec{\Im}\begin{pmatrix}f\left(x,y,z\right)\\ g\left(x,y,z\right)\\ h\left(x,y,z\right)\end{pmatrix}=\begin{pmatrix}ky-v_1x+yz\\ kx-v_2y-xz\\ -v_3z+xy\end{pmatrix}\quad(14)$$

where the functions *f*, *g* and *h* are infinitely differentiable with respect to all x_i, and *t*, i.e., are C^∞ functions in a compact E included in $\mathbb{R}^3$ and with values in $\mathbb{R}$.

If we consider that all parameters are null we have a homogeneous differential system, the location of the points where the local torsion (5) vanishes leads to the following equation:

$$\phi\left(\vec{X}\right)=\left(x-z\right)\left(x+z\right)\left(y^2+z^2\right)\left(x^2+y^2\right) \quad (15)$$

The equation for the inflection curve, or the first extactic curve (9) provides exactly the same result. Moreover, the Lie derivative of ϕ leads to $L_{\vec{X}}\phi=Tr\left[J\right]\phi\left(\vec{X}\right)$, i.e.,

$$L_{\vec{X}}\phi=0 \quad (16)$$

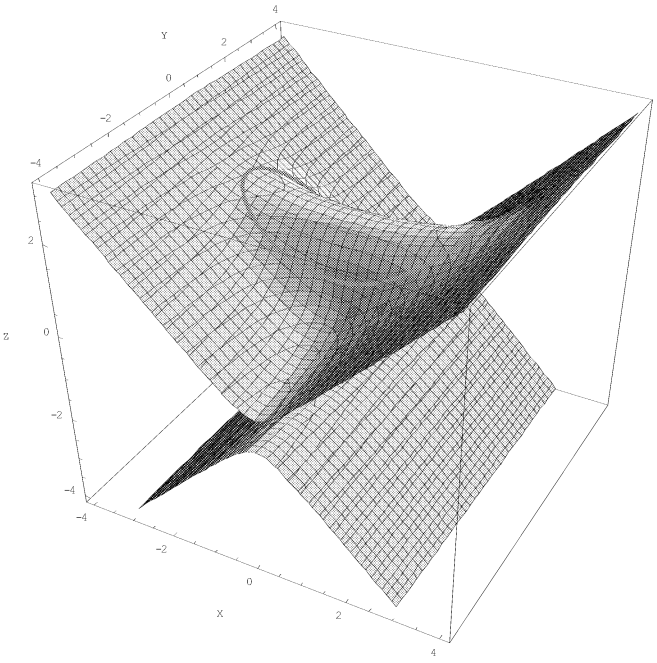
since $Tr\left[J\right]=-\left(a+b+c\right)$. So, if all parameters are null ϕ is first integral of the Pikovskii-Rabinovich system (14).

Note.

If parameter *k* is not null system (14) is not a homogeneous differential system anymore. Nevertheless, the location of the points where the local torsion (5) vanishes as well as the equation for the inflection curve, or the first extactic curve (9) leads to the following equation:

$$\phi\left(\vec{X}\right)=\left(y^2+\left(k-z\right)^2\right)\left(k-x+z\right)\left(k+x+z\right) \quad (17)$$

Moreover, the Lie derivative of ϕ still vanishes. Then, ϕ is first integral of the Pikovskii-Rabinovich system (14).



Pikovskii-Rabinovich system for all parameters null except *k*=1.

CONCLUSION

In this work, a new approach for directly detecting invariant algebraic manifolds of degree 1 of differential systems has been presented. Identity with both existing methods of Lagutinskii and Darboux has been established. Then, application to nonlinear dynamical systems of dimension two and three has shown under conditions that invariant manifolds play an important role in the dynamic of such dissipative systems since they enable to find first integral even to build general integral of such systems. Moreover, the last note highlights the fact that Darboux theory of invariant curves (resp. surfaces) may be applied to non homogeneous differential systems.

ACKNOWLEDGEMENTS

Authors would like to thank Professors M. Jaume Llibre and Hector Giacomini for their useful collaboration.

REFERENCES

Christopher, C. Llibre, J. and Pereira, J.V. 2007. "Multiplicity of invariant algebraic curves in polynomial vector fields". Pacific J. of Math., Vol. 229, (1), 63-117.

Coddington, E.A. & Levinson., N., 1955. *Theory of Ordinary Differential Equations*, Mac Graw Hill, New York.

Darboux, G. 1878. "Mémoire sur les équations différentielles algébriques du premier ordre et du premier degré". Bull. Sci. Math. Sér. 2 (2), 60-96, 123-143, 151-200.

Demazure, M. 1989. *Catastrophes et Bifurcations*, Ellipses, Paris.

Dobrovol'skii, V.A., Lokot', N. V. and Strelcyn, J.-M, 1998. "Mikhail Nikolaevich Lagutinskii (1871–1915), un mathématicien méconnu". Historia Mathematica, Vol. 25 (20), 245-264.

Lagutinskii, M. N., 1908. Partial Algebraic Integrals. Kharkov, in Russian.

Lotka, A. 1925 Elements of Physical Biology, Dover Publications, New York.

Pikovskii, A.S., Rabinovich, M. I., V. Y. Takhtengerts, 1978. "Onset of stochasticity in decay confinement of parametric instability". *Soc. Phys. J. E. T. P.*, t. 47, 715-719.

Volterra, V., 1931, *Leçons sur la théorie mathématique de la lutte pour la vie*, Gauthier-Villars, Paris.

JEAN MARC GINOUX was born in Toulon, France and went to the University of Nice Sophia-Antipolis, at I.N.L.N. of Nice and then at the University of South, where he studied nonlinear and chaotic dynamical systems with Professor Bruno Rossetto. He obtained his Ph-D in Applied Mathematics in 2005. He worked for a couple of years for the Department of Education before moving in 2004 to I.U.T. of Toulon where he has been working in Dynamical Systems ever since.

ON THE DECENTRALIZED DYNAMIC GRAPH COLORING PROBLEM

Antoine Dutot, Frédéric Guinand, Damien Olivier and Yoann Pigné
LITIS - University of Le Havre
25 rue Ph. Leblon - BP 540
76058 Le Havre Cedex - France
Email: Surname.Name@univ-lehavre.fr

INTRODUCTION

The central topic of this contribution is the conception of decentralized methods for dynamic graphs processing. The discussion is illustrated by the well-known graph coloring problem. In the wide domain of Complex Systems, graphs are omnipresent. Above all, this is due to the nature of these systems. Indeed, the emergent properties of complex systems arise from the dynamics of the interactions between entities belonging to the system and between these entities and the environment. Domains so different as, Biology, Physics, Social Sciences or Computer Science share these notions. If anchovy shoals, starling flocks, people crowds or mobile ad hoc networks are under study, the same universal model may be considered: dynamic graphs. Each entity, whatever its type, person, animal or machine, is autonomous and interact locally with other entities located in its neighborhood of communication. The communication may be visual, oral or of another nature like electromagnetic waves for instance. However, not all computer networks are comparable with other kinds of interaction networks. We think that DT-MANETs, Delay-Tolerant Mobile Ad-Hoc Networks are especially close to natural interaction networks. Such computer networks are made of mobile communicating devices that may be switched on or off at any moment and in a sudden way, and during its evolution this network may also be partitionned as it may be the case with people crowds in human societies or birds flocks. Whatever the considered domain, the main sources of dynamics are: the volatility of entities and the dynamic of the topology of the interaction network. The volatility of entities may have many differents reasons. A machine may be subject to a failure or may have been voluntarily switched off/on for energy saving for instance. Biological entities may appear and disappear according to life cycle (death/birth), or may be disconnected from their neighborhood for a while (sleep). Then, the obtention as a side-effect, by a decentralized method, of global-range structures in dynamic graphs, seems to raise from the same mechanisms aa emergent properties in natural complex systems. And we hoppe that a better understanding of such kind of mobile networks may lead to a better understanding basic mechanisms of emergence in natural complex systems.

In the context of dynamic graphs, the conception of decentralized algorithms for reaching a particular goal (broadcasting, routing, counting, stations coupling, search for particular paths, etc) prove to be of another nature that in a static setting. Indeed, in a dynamic framework, the algorithm has not only to build a structure but it also has to maintain this structure. So the algorithm itself has to behave in a dynamic way. This situation occur also in distributed systems, when some tasks like file transfert or dynamic scheduling have to be performed in a continuous way. Some of these tasks correspond to the classical graph coloring problem. Within this paper, a self-organized approach is described and its performances in term of graph coloring are compared with those obtained using classical static algorithms like Welsh and Powell algorithm. These algorithms are statically applied to the whole graph between to steps of dynamic events.

GRAPH COLORING PROBLEM

The aim of the graph coloring problem consists in using as few colors as possible such that two adjacent vertices own a different color. A formulation of the k-colorability problem might be found in [GJ79]:

> **Problem Name :** *Graph k-coloring.*
>
> **Parameter Description :** a colored graph $G = (V, E, C)$, such that V denotes the set of vertices of the graph, E the set of edges belonging to the graph and C the colors associated to vertices.
>
> **Question :** Is it possible to color the graph using only k colors such that two adjacent vertices do not share the same color ?

This problem formulated in a static context has been proved to be NP-complete in 1972 by Richard Karp. For dynamic graphs, this problem can be formulated in a very similar way, except that time appears in the description:

> **Problem Name :** *Dynamic graph coloring.*
>
> **Parameter Description :** a dynamic colored graph $G(t) = (V(t), E(t), C(t))$, such that $V(t)$ denotes the vertices of the graph at time t, $E(t)$ the edges belonging to the graph at time t and $C(t)$ the colors of the vertices at time t.
>
> **Question :** Is it possible to color the graph using only $k(t)$ colors such that at t two adjacent vertices do not have the same color ?

Of course, as the static version of the problem is NP-complete, so is the dynamic one.

We are willing to solve this problem using a decentralized approach. Indeed, we would like the applicability of the method to be as wide as possible. For instance, if we consider this problem in the context of mobile ad hoc networks (MANETs), no device has a global view of the network and only strategies relying on local information are applicable.

For summing up, we address the problem of coloring a graph, that may change during the time it is considered, using as few colors as possible, by means of a decentralized strategy. In addition, we want to maintain the coloring property at any time in the graph, that is, no two adjacent nodes may have the same color at the same time. We consider that two vertices are connected as soon as at least one station is aware of the existence of the other.

Dynamic Decentralized Graph Coloring

Hypotheses

When dealing with decentralized approaches for graph coloring, there exist, at least, two ways of performing the task. On the one hand, one node initiates the process, on the other hand, the process starts on every node at a given date t_{start}, or as soon as the node is created, or appears in the network. In the sequel, we will refer to *broadcasting coloring* for the first scenario, and to *generalized coloring* to the second one. In all cases, it is supposed that:

1. no color set is initially known by the nodes,
2. colors are not comparable, that is there is no order on the set of colors,
3. when a node needs to use another color, this color is obtained by a random process that may produce either a new color or a color already existing,
4. finally, each vertex knows its neighborhood and may access to the color of his neighbors.

In the broadcasting coloring, one node begins with a new color and broadcast this color to his neighbors. Then, asynchronously, neighbors perform a coloring of themselves by either creating a new color or by choosing in the received color tab an existing color that do not belong to their neighbors. In a static context, each node may be processed only once, but in a dynamic setting the neighborhood of a vertex may change during the process such that the constraints are no more respected. So, each vertex should be both proactive and reactive. Proactivity is necessary for involving new nodes in the process, and reactivity is required for verifying the coloring constraint. In the generalized coloring scenario, each node comes with its own color. There is a high probability that the coloring constraint is verified at the beginning. The goal of each node, while preserving the coloring constraint, consists in trying to switch to another color already used in the network in order to reduce the number of used colors.

Algorithm

Here is the proposed algorithm. It is executed by each vertex.

```
while true do
    if invitation to start is received from a neighbor then
      coloring()
    endIf
    if change in neighborhood detected then
      coloring()
    endIf
    if request for myColor then
      send myColor
    endIf
    if request for my color tab then
      send AllColors[]
    endIf
endWhile


coloring()
    Impossible[] ← request neighbors to their color
    AllColors[] ← request neighbors to their color tables
    if there is a color c such that c ∈ AllColors
    and c ∉ Impossible then
      myColor ← c
    else
      AllColors[] ← AllColors[] ⋃ new color c
      myColor ← c
    endIf
```

Experiments

Preliminary results are reported in this section. The experiments were led using GraphStream, a library for visualizing and manipulating dynamic graphs [DGOP07].

First we note that the dynamic decentralized algorithm can self-adapt to the situation since it is able to increase and decrease the number of colors according to the dynamic of the graph.

Second, it appears that the performances in terms of number of used colors of the dynamic decentralized algorithm is almost as good as the Welsh and Powell algorithm, as it can be seen on the pictures 1 and 2.

References

[DGOP07] Antoine Dutot, Frédéric Guinand, Damien Olivier, and Yoann Pigné. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. In *EPNACS'2007 workshop of ECCS'07*, 2007.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman, New York, 1979.

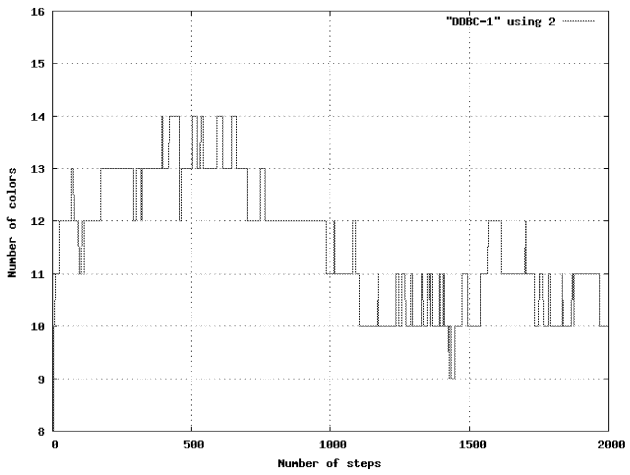


Figure 1: Results obtained using the Dynamic Decentralized Broadcasting Coloring algorithm. We can observe that the number of used colors in the graph may increase for constraint satisfaction, but it may also decrease according to the dynamic of the graph.

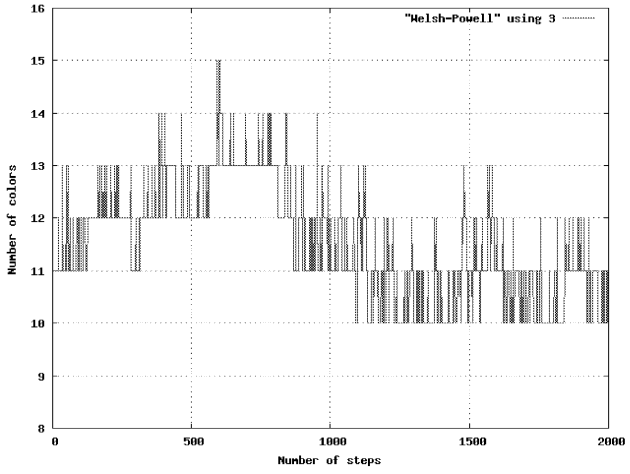


Figure 2: Results obtained for the same series of graphs by applying the Welsh and Powell static algorithm at each time step.

Control on System Diffusion Using Genetic Automata

Luaï Jaff[(1)], Cyrille Bertelle[(1)], Gérard H.E. Duchamp[(2)]

[(1)]LITIS - University of Le Havre, 25 rue Philippe Lebon, BP 540, 76058 Le Havre Cedex, France
email: luai.jaff@gmail.com, cyrille.bertelle@litislab.eu

[(2)]LIPN - University of Paris XIII, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France
email: ghed@lipn.univ-paris13.fr

KEYWORDS
Self-organization; Genetic Automata; Diffusion; Control

ABSTRACT

Complex systems (LeMoigne 1999) are models which describe dynamic organizations or systems where the internal interaction network between their components does not allow the control of individual components with efficience. In a lot of cases, the individual control leads to break the whole structure itself. In other cases, some complex systems exhibit resilience properties which lead them to recover their initial state after small perturbations or individual controls. Controlling complex systems need to not trying to manage the individual controls but to manage the global control, guiding the whole system without directly act on individuals themselves. The study presented in this paper explains how to control a high level system property, the diffusive behavior, by spectral analysis over genetic automata. First experiments are presented using MuPad implementation.

COMPLEX SYSTEMS AND FEED-BACKS FOR ECONOMY

We describe the systemic approach of economical system modelling, especially the feed-back concept which can be considered as a global control over the system components.

Complexity in Economy

We can classify the economical systems in two main classes (Marshall 1920):

- The classical approaches based on a rational choice of sets of equations representing the state of the economy. The drawbacks from this approach are: the unrealistic assumption of rational choices and perfect information; assumptions on homogeneity and lack of distinction between the agent and the aggregate level; inability to account for the emergence of new kinds of entities, patterns, structures, etc.
- Complexity perspective approach. This approach is more recent as a view of economy. It focusses on viewing the economy as an evolving complex system, using Holland's complex adaptive systems paradigm (Holland 1975).

With the complexity approach, the aim is to understand how global economic phenomena arise purely from the local interactions and local knowledge of the agents. This stems from the recognition that there is no central or global control in an economy and that the global regularities which arise are purely due to the local interactions of adaptive autonomous agents. The agents within an economic network may be individuals or institutions. We can consider an institution agent as being composed of many individual agents.

Feed-Backs Modelling

W. Brian Arthur (Arthur 1999), and other economic theorists have been interested to a view of the economy based on feedbacks, more precisely, positive feedback mechanisms. We can say that modern economies can be divided into two interrelated branches :

- Diminishing Returns and Negative Feedback.

  Diminishing returns refers to the notion that the return that a company receives for additional effort decreases as the number of units (outputs) increases. Diminishing returns explains why industrial companies become more inefficient once they grow over a certain size.

  Negative feedback is the process of feeding back to the input a part of a system's output, so as to reverse the direction of change of the output. This tends to keep the output from changing, so it is stabilizing and attempts to maintain constant functionning conditions.

Diminishing returns assumptions are the basis of the conventional economic theory. The actions of the economy induces a negative feedback which leads to predictable equilibrium for prices and market share. Negative feedback stabilizes the economy because any major change will be cancelled out by the reaction they generate. Diminishing returns imply a unique equilibrium point for the economy.

Alfred Marshall (Marshall 1920) believed that we can not applied the increasing returns everywhere in the economy. His remark leads W. Brian Arthur to observe that the part of the economy that are resource-based (the traditional part) are still subject to diminishing returns, on one hand. And the part of the economy that are knowledge-based (the newer part) are still subject to increasing returns, on the other hand as defined in the following.

- Increasing Returns and Positive Feedback

  Increasing returns refer to the notion that the greater the size of the network, the greater the advantage of each participant of the network (network effects). Each participant of the network brings value to the overall network. This is in contrast to diminishing returns which refers to the greater the size (number of users) the less each participant can benefit from participation. Increasing returns generate instability.

  We call a feedback mechanism positive if the resulting action goes in the same direction as the condition that triggers it. Positive feedback is an open system contain many types of regulatory systems, among which are systems that involve positive feedback and its relative negative feedback.

  Increasing returns make for multiple solutions (multiple equilibrium points), and it generate instability or criticality stability following Per Bak (Bak 1996).

After describing the scientific environment of complexity, we give in the next section formal tools allowing to model some auto-regulation processes leading to implement the feed-back mechanisms.

AUTOMATA-BASED SYSTEM MODELLING

We focus our attention on a special kind of automata with outputs which are efficient in operational way (Hopcroft et al. 2001). This automata with output are called automata with multiplicities and they are defined in the following.

Automata with multiplicites

An automaton with multiplicities is based on the fact that the output data of the automata with output belong to a specific algebraic structure, a semi-ring. In that way, we will be able to build effective operations on such automata, using the power of the algebraic structures of the output data. And we are also able to describe this automata in matrix representation with all the power of the linear algebra.

Definition 0.1 *An* **automaton with multiplicities** *over an alphabet* $\sum$ *and a semi-ring* K *is the 5-uple* Σ, Q, I, T, δ *where*

- $Q = \{S1, S2, ..., Sn\}$ *is the finite set of state*
- $I : Q \to K$ *is a function over the set of initial states, which associates to each initial state a value of K, called entry cost, and to non-initial state a null value*
- $T : Q \to K$ *is a function over the set of the final states, which is associated to each final state a value of K, called final cost, and to non-final state a null value*
- δ *is the transition function, that is* $\delta : Q \times \Sigma \times Q \to K$ *which from a state S1 , a letter a and a state S2 go to a value z of K if it exist a transition labeled with a from the state S1 to the state S2 and with the output z.*

We remark that

- automata with multiplicities is a generalization of finite automata. In fact, finite automata can be considered as automata with multiplicities with for the semi-ring K, the boolean set $B = \{0, 1\}$. To each transition we affect 1 if it exists and 0 if not.
- We have not yet, on purpose, defined what a semi-ring is. Roughly it is the least structure $(K, +, x)$ that allows matrix computation with units (one can think of a ring without the **minus** operation)

Linear Representation

The previous automata with multiplicities can be expressed by a linear representation which is a triplet

$$p = (\lambda, \mu, \gamma)$$

with $\lambda \in K^{1 \times n}$ is a row-vector which coefficients are $\lambda_i = I(S_i)$, $\gamma \in K^{n \times 1}$ is a row-vector which coefficients are $\gamma_i = T(S_i)$, and $\mu : \Sigma \to K^{n \times n}$ is a morphism of monoids such that $\forall a \in \Sigma$, the coefficient on the i-th row and j-th column of all transitions labeled with a is $\mu(a)_{ij} = \delta(S_i, A, S_j)$.

In the following, we describe the linear representation of the automata with multiplicities which corresponds to the the following figure.

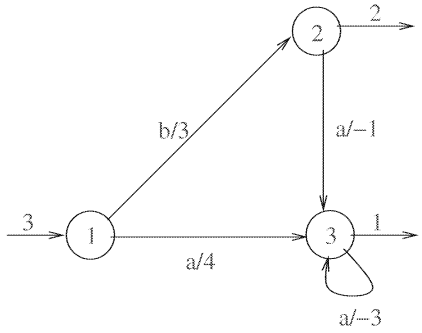


Figure 1: Automata with multiplicities

In this figure, the input states are characterized with input arrows from nothing and the output states are characterized with output arrow to nothing. On each transition, we indicate before the semi-column, the input data and after the semi-column, the output associated for each input data.

The linear representation is the following

$$\lambda = \begin{pmatrix} 3 & 0 & 0 \end{pmatrix}$$

$$\gamma^t = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix}$$

$$\mu(a) = \begin{pmatrix} 0 & 0 & 4 \\ 0 & 0 & -1 \\ 0 & 0 & -3 \end{pmatrix}$$

$$\mu(b) = \begin{pmatrix} 0 & 3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

GENETIC AUTOMATA

Genetic algorithms play an important role in the studies of complex adaptive systems, ranging from adaptive agents in economic theory to the use of machine learning techniques. We describe the principles of these algorithms and then their application to genetic automata.

Genetic Algorithms and Operations

The major aspect of genetic algorithms is the adaptation property. This property can be considered as the search of the optimum of a specific function. Genetic algorithms work on a population of individuals. The individuals are represented with chromosomes which are composed with primitive information called alleles. In term of computable formalization, chromosomes are generally strings or sequences of information over a finite alphabet. To begin the algorithm, we generate a population of chromosomes.

The algorithm is based on a variation-selection process:

- The variation step concerns the basic genetic operators on the individual level and so this step acts on the chromosomes. These basic operators are composed of reproduction/duplication, crossing-over and mutation and we describe how implement them for genetic automata in the following section. The result gives another population with a greater number of chromosomes than the initial one
- The selection step concerns the population level inside which a selection function modifies the whole population constitution. This step leads to keep only some of the chromosomes that have been generated during the variation step and which satisfy to specific constraints

Automata Chromosome Coding and Genetic Operators

We define the chromosome for each automata with multiplicities as the sequence of all the matrices associated to each letter from the (linearly ordered) alphabet. The chromosomes are composed with alleles which are here the lines of the matrices.

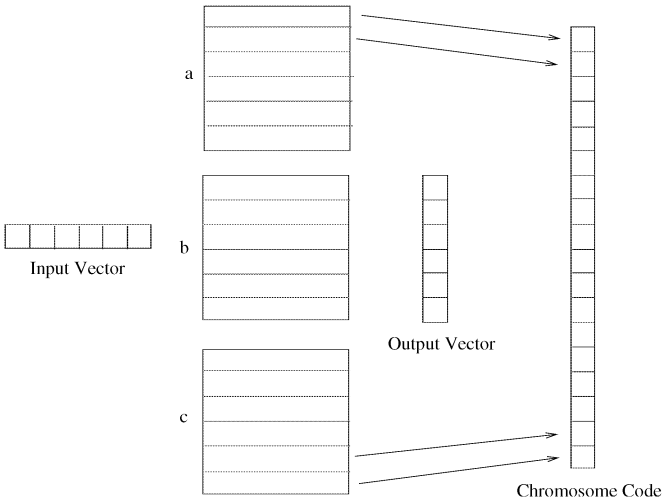


Figure 2: Chromosome code

In the processus, genetic algorithms generate new automata containing possibly new transitions from the

ones included in the initial automata. The genetic operators are applied as described in the following:

- Reproduction/duplication: this operator simply copy the automaton chromosome.
- Crossing-over: From 2 chromosomes, it consists in cutting of the initial chromosomes in the same place chosen at random and in permuting the two parts of the matrices lines chains.
- Mutation : In this step, we chose a few chromosomes at random to be candidate for the mutation process. If selected, we modify one line of the matrix lines chromosome. If the matrix lines have to respect some properties (like probabilistic matrix) we constraint the new values to respect them.

EXPERIMENTS ON DIFFUSIVE PROPERTY CONTROL

We present in the following some experiments based on the package MuPAD-Combinat which implements many data structures related to combinatorics (Hivert and Thierry 2004). The genetic operators are developped on the predefined domain called Weighted Automaton from this package.

Our experiment consists to define chromosome code for probablitistic automata which are specific automata with multiplicities satisfying the condition that the sum of each lines coefficient must be equal to 1.

Fitness Function and Auto-Regulation Process Based on Spectral Analysis

We define a population of genetic probabilistic automata generated at random and we compute for each matrix, the secondary greatest eigenvalue module (the first one is always equal to 1, because of the probablitistic matrices used, respecting Perron-Frobenius Theorem). The genetic algorithm is described in 1.

Experiment on non Diffusive Systems Control

In the figure 3, we represent a series of histograms computed at different steps where we indicate the number of automata according to their secondary greatest eigenvalue module. With the evolution of the population, we observe that the selection will generate population with chromosome of high secondary greatest eigenvalue module.The x-scale is adaptive to the interval containing all the values of this secondary greatest eignevalue for the automata population.

Algorithm 1: Auto-regulation algorithm for genetic automata population

```
Generation of P the initial population of probabilistic automata chromosome;
while not end of iterations do
    for each chromosome c ∈ P do
        Duplicate c;
        Update P;
    for each couple of chromosomes (c1, c2) ∈ P do
        Make crossing-over on (c1, c2);
        Make mutation on c1 and c2 with low random probability;
        Update P;
    for each chromosome c ∈ P do
        Compute the fitness of c as the secondary greatest eigenvalue module;
    Select from P the half higher fitness population and update P with this selection;
```

Results analysis and interpretation

As expected by the fitness function, we generate successive populations which are characterized by the convergence process which leads to obtain a population where the greatest secondary eigenvalue module of all automata with multiplicities is equal 1.

This processus is an auto-regulation processus on the population itself, managed by a feed-back of the result control on the system itself.

By controlling the generating automata population with eigenvalues module equals 1, we constraint the system to not be dissipative. The spectral analysis and the eigenvalues module give some indications on the dissipative aspect during the evolution. If the eigenvalue modules are less than 1, then we can conclude that a dissipation exists.

The non dissipative evolution process over automata with multiplicities proposed here, is a tool based on the power of the algebraic representation of the automata with multiplicities, using some computation on their matrix representation. So we deal on a dynamic data structure according to the complexity of dissipative process over some self-organized systems. Without managing each matrix coefficient value, we give a kind of control tool according to dissipative process increasing or decreasing.

CONCLUSION AND PERSPECTIVES

In this paper, we present a methodology which allows to control a complex system by a global function which is based on the use of a genetic probabilistic automata population. We scan this population by means of a spectral analysis and control the system by reducing its diffusive properties. The applications to economy can be relevant and we propose by this method, a practical

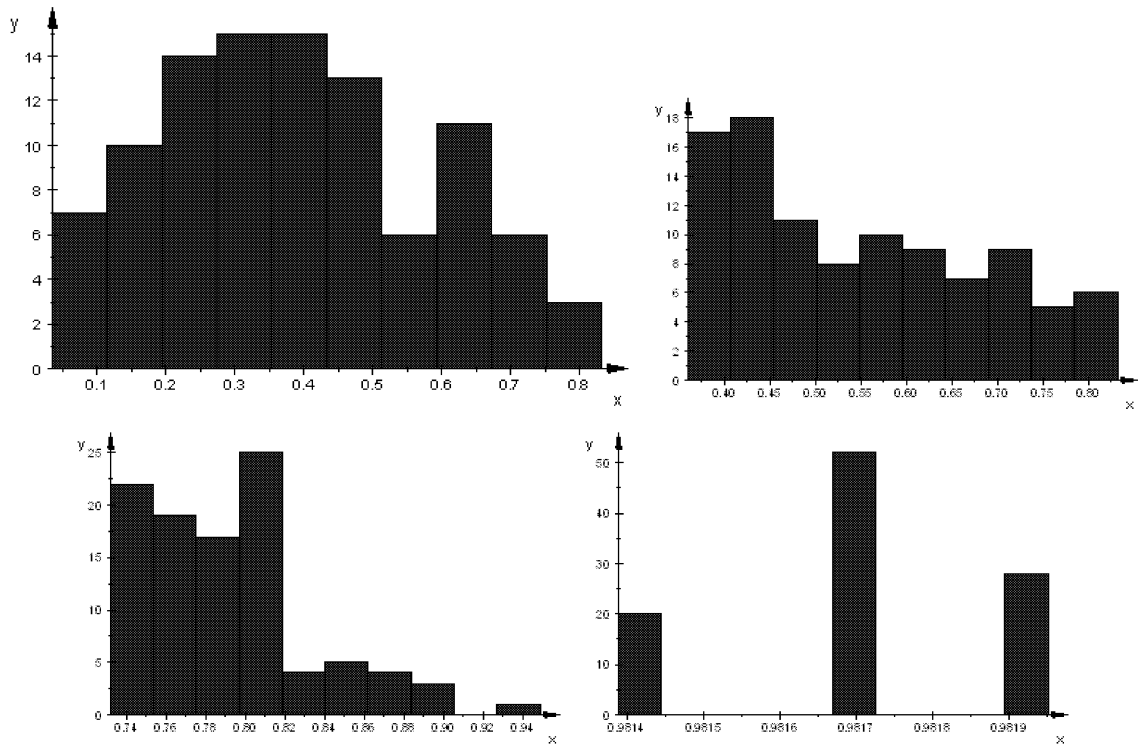


Figure 3: Histograms of the evolution of automata population according to their secondary greatest eignevalue module at iteration 1, 10, 20, 30 and 40

and computable way to implement the feed-back mechanisms over systems in economy.

REFERENCES

Arthur B., 1999. *Complexity and the Economy. Science*, 284, 107–109.

Bak P., 1996. *How nature works - the science of self-organized criticality.* Springer Verlag.

Hivert F. and Thierry N.M., 2004. *MuPAD - Combinat, an open-source package for research in algebraic combinatorics. Seminaire Lotharingien de Combinatoire*, 51.

Holland J., 1975. *Hidden Order - How adaptation builds complexity.* Helix Book.

Hopcroft J.; Motwani R.; and Ullman J., 2001. *Introduction to automata theory, languages and computation.* Addison-Wesley.

Jaff L., 2007. *Dynamic Data Structures for Complex Systems.* Ph.D. thesis, University of Le Havre.

Kline D., 2001. *Positive feedback, lock-in and environmental policy. Policy Sciences*, 34, 95–107.

LeMoigne J.L., 1999. *La modélisation des systèmes complexes.* Dunod.

Marshall A., 1920. *Principles of Economics.* MacMillan.

Schutzenberger M., 1961. *On the definition of a family of automata. Information and Control*, 4, 245–270.

Building upon Fast Multipole Methods to detect and model organizations

Pierrick Tranouez and Antoine Dutot
LITIS
Université du Havre
UFR Sciences et Techniques
25 rue Ph. Lebon - BP 540
76058 Le Havre Cedex – France
{Pierrick.Tranouez, Antoine.Dutot}@litislab.eu

KEYWORDS

Organization detection, structure management, multiscale, N body systems, hierarchical tree code approximations

ABSTRACT

Many models in natural and social sciences are comprised of sets of interacting entities whose intensity of interaction decreases with distance. This often leads to structures of interest in these models composed of dense packs of entities. Fast Multipole Methods are a family of methods developed to help with the calculation of a number of computable models such as described above. We propose a method that builds upon FMM to detect and model the dense structures of these systems.

INTRODUCTION

We study in this paper dynamic systems composed of many interacting entities. We are interested in their modeling, simulation, in the way they evolve and organize themselves.

Among these theoretical systems, a class of them is used as models for different natural science systems. In this class, systems verify the following properties:

1. Entities follow an analytical model
2. Each entity interacts with all the others.
3. The strength of the interactions decreases with the distance separating the entities implied in the relationship

It is to this class that N-Body problems in physics belong to for example.

Property 2 makes computing a simulation through the iterative calculations of the model in 1 costly, in $O(n^2)$ for n entities. There are mathematical techniques to accelerate this computing, such as Fast Fourier Transform or Fast Multipole Methods. We will deal in this paper with hierarchical methods such as FMM.

Furthermore, systems of the class described above have also some remarkable properties in their organizations. As interactions intensity decrease with the distance and these interactions are often of an attractive nature, aggregate or clusters often emerge of their simulation, and play important parts as structures and organizations of the simulation.

We give in this article a method that detects and manages such organizations in this class of models, in connection with an FMM simulation (or hierarchical tree code approximations in general).

KERNELS AND FAST MULTIPOLE METHODS

The Introduction may give the impression that our organization-handling scheme works in only very restricted cases, but systems verifying these properties often appear in scientific literature [Beatson and Greengard].

Physics examples

Provided N bodies of mass m_i a description of the gravitational field in x_j is Newton law [Newton 1686]:

$$g(x_j) = \sum_{\substack{i=1 \\ i \neq j}}^{N} \mathcal{G} m_i \frac{x_i - x_j}{\|x_j - x_i\|^3}$$

Where $\mathcal{G}$ is the gravitational constant.

Similarly for N bodies of charge q, Coulomb law [Coulomb 1785] describes the electrostatic field in x_j:

$$E(x_j) = \sum_{\substack{i=1 \\ i \neq j}}^{N} \mathcal{K} m_i \frac{x_i - x_j}{\|x_j - x_i\|^3}$$

Where $\mathcal{K}$ is the electrostatic constant.

In both of these, each body interacts with all the other, with an intensity that is inversely linear to the square of the distance. It is the same for Biot-Savart law, relative to both magnetism (under) and fluid dynamics [Leonard 1980]:

$$B(x) = \frac{1}{c} \int J(y) \frac{x - y}{|x - y|^3} dy$$

Where $J(y)$ is the current density in y.

The same can be written for diffusion, of heat for instance.

For all of these, structures of close entities play an important part. In gravitation, on different time scale they are the galaxies, solar systems, stars, planets. In magnetism,

hydrodynamics and aerodynamics, congruence leads to vortexes or coils of different sizes, ranging from turbulence to the Gulf Stream or the red spot of Jupiter. In the Sun, congruence of plasma matter lead to sunspots and solar flare, and combined with fluids and gravity to coronal mass ejections.

Life sciences examples

Many living systems can be described the same way [Simon 1996] [Frontier and Pichot-Viale 1998], with hierarchical spatial organizations. One of these models is the boids [Reynolds 1987]. This is the model we will use to exemplify our method, so we will describe it in some depth.

Boids are entities moving in a 2D or 3D space. They are ruled by quite simple rules, *Cohesion, Separation* and *Alignment*. Cohesion describes how boids try to fly in the direction of surrounding boids. Separation describes how boids are repulsed by other boids if these are really close. Alignment describes how boids try to fly in the same direction as the boids that environs them. A more formal description is:

Let j be a boid situated in the point x_j of an affine space. Let $\eta(j)$ be its neighboring space, ie the set of boids not further from j than a certain limit at a given time t, excluding j itself.
Let v_t^j be the speed of j at time t. All the x_k below should be written x_t^k but are conventionally simplified to easy up the reading.

We define:

$$c_{t+1}^j = \sum_{i \in \eta(j)} \frac{x_i}{\|x_i - x_j\|^2} - x_j$$

$$s_{t+1}^j = \sum_{i \in \eta(j)} \frac{(x_j - x_i)}{\|x_i - x_j\|^3}$$

$$a_{t+1}^j = \sum_{i \in \eta(j)} \frac{v_t^i}{card(\eta(j))\|x_i - x_j\|^2}$$

Which we combine:

$$v_{t+1}^j = \alpha v_t^j + \beta c_{t+1}^j + \gamma s_{t+1}^j + \delta a_{t+1}^j$$

Where $\alpha, \beta, \gamma, \delta$ are coefficients used to balance the different components of the speed.

This is for one species of boids. If different species of boids are considered, another stronger inter-species Separation is furthermore implemented to represent a tendancy to flock to their species and flee others [Dutot 2005]. Each species member speed is only influenced by its co-members as far as Alignment, Cohesion and intra species Separation are concerned. They can also have each species their own coefficients α, β, γ and δ.

Hierarchical methods to accelerate the computing

The computing of the relation above is of a complexity of the order of the square of the number of entities/bodies involved. This can be costly as the size of the simulation increases.

An often used technique to improve this is is to recursively divide the simulation space in a tree. The root of the tree encompasses the whole space. Its children are regular subdivision of the space, for example in 4 equal surfaces. The process is repeated recursively. We use an adaptive subdivision, which means regions of space containing many entities are more divided that sparser ones. The entities are associated to the leaves of the tree, which is adapted at each simulation step.

The degenerated kernel relations described above are then computed. For that, the contribution of close entities is added exactly, while the contribution of further entities is averaged at the level of the leaf it belongs to. The exact economy is difficult to compute precisely as it depends on the structure of the tree, but it can be brought back to an average of $O(NlogN)$ or even $O(N)$ [Beatson and Greengard] for N entities in the simulation. Furthermore, the loss of precision can be bound.

We originally used FMM in [Tranouez 2005] in a fluid flow simulation [Tranouez et al. 2005]. We discovered at the time that the tree used for the poles visually summarized the organizations appearing during the simulation (groups of vortexes of different shapes and sizes in this case). We therefore created an algorithm that formalized this visual discovery. We will now describe this method, on a boids application.

THE METHOD

Relationship between *n*-trees and organizations

As said above, the hierarchical subdivision of space can be represented by a tree whose root is the whole considered space and branches and leaves are subdivisions of this space. The space is subdivided so that there is a constant, predefined, maximum number of entities (which, per analogy with the N-body problem, we will call *bodies)* in each subspace.

We call *cell* a node of the tree and its associated subspace. Depending on the way the space is decomposed, the number of sub cells varies. Often 2D spaces are cut in four cells yielding *4*-trees or "quadtrees", and 3D spaces are decomposed in eight cells, producing *8*-trees or "octrees".

The number of bodies per leaf cell is variable. Some methods can put only one body per cell, others put several bodies in one cell. Bodies are only "present" in the leaf spaces, and super cells aggregate the data of their sub cells.

It is necessary to subdivide a cell when the maximum number of bodies per cell is reached, and *n* cells merge into one when there is less than the maximum number of bodies

per cell in all the *n* cells. Therefore, this subdivision is adaptive, and changes during time.

We call a "level" all the cells of the tree at a given depth in the tree. The nature of the subdivision yields many subdivided cells where there are many bodies. The intuition of the method is that it is easy to identify dense groups of bodies (that is places where interactions are stronger) by looking at a representation of the subdivided space (figure 1).

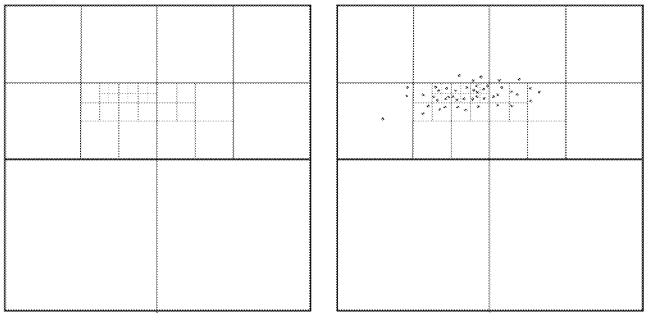


Figure 1 : The space subdivision, with and without bodies. Each cell contains a maximum of 3 bodies.

Finding organizations with the n-tree

Our method relies on the use of the *n*-tree to detect organizations, therefore sharing the use of this tree both for the FMM and for the organization detection.

The main idea is to consider only the lower levels of the tree as shown on figures 2 and 3, and then to cut this set of cells in groups of adjacent cells. The tree in figure 3 represents the subdivided space of the figure 2. There are at most three bodies per cell here. The search for organizations begins by considering only the lower levels of the tree. That is, cutting the bottom of the tree (here under the horizontal dotted line). In this restricted set of cells, we remove all cells that do not contain bodies (in grey on the figure). Notice that this selects only "leaf" cells. This set must then be cut in groups of adjacent cells. These groups can be considered a good approximation of searched organizations.

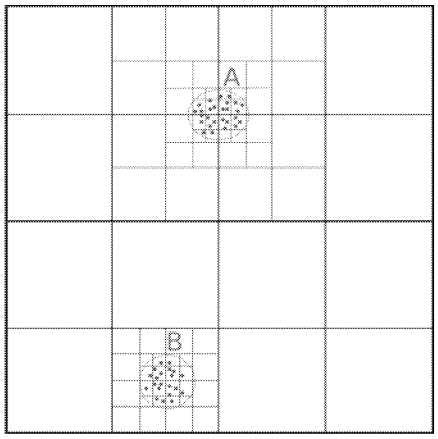


Figure 2 : Two organizations A and B and the subdivided space

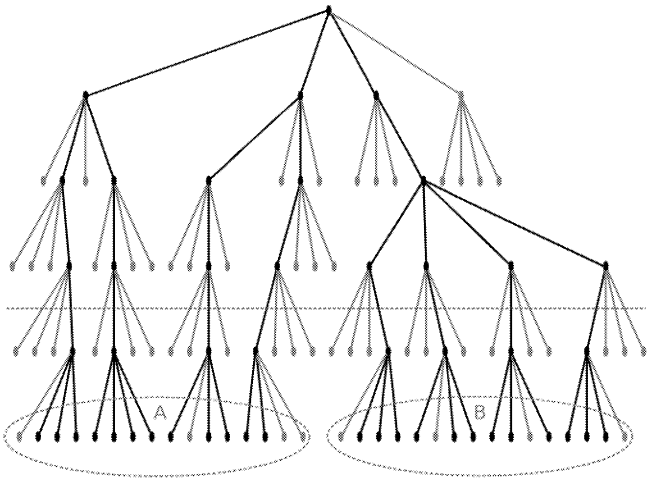


Figure 3 : The A and B organizations in the n-tree

In two dimensions, one can also see the subdivisions of space as being higher in a third dimension. The more space is subdivided, the higher a point in space is. As subdivision matches body density, the denser the bodies are the higher is their third coordinate. The idea is then to cut this representation and consider only the cut parts to search for dense groups of bodies, as shown on Figure 4.

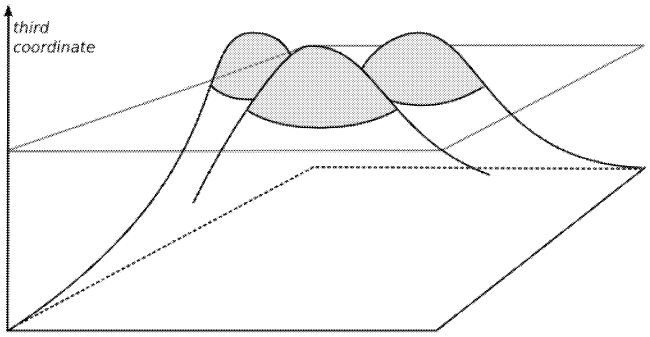


Figure 4 : Conceptual representation of the method.

Algorithm

The organization detection algorithm does not work directly on the n-tree, but on a set of cells C corresponding to some levels of the n-tree. They correspond to all the cells of the tree that are deeper than a given depth d.

In this set, we keep only the leaf cells, as only these cells "contain" bodies.

The algorithm will then try to cut this set in groups of adjacent cells. We will consider these groups as organizations. Adjacent cells are cells that share a face, an edge or a point (Figure 5).

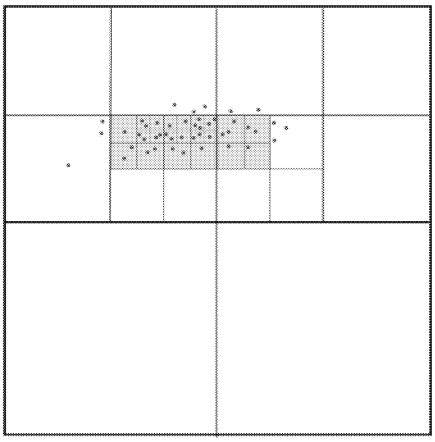


Figure 5 : Adjacent cells delimit organizations

The creation of the groups can be done easily with a $O(n^2)$ algorithm, with n the number of cells in C. Consider one cell c taken at random in C and remove it from C. Consider all the other cells of *C* and check if they share a face, edge or point with c. If they do, put them in the same group and remove them from C.

When there are no more cells of C adjacent to c, and if *C* still contains cells, create another group, choose randomly another cell c' and restart the procedure. Do this until C is empty.

```
Algorithm GroupCells( C ) returns G
C: set of cells
G: set of sets of cells
c: cell
g: set of cells
While not C.isEmpty do
     g <- new group
     c <- C.removeRandom
     G.add( g )
     g.add( c )
     ForAll cell d in C do
          If d.shareFaceOrEdgeOrPoint( g )
          then
               C.remove( d )
               g.add( d )
          EndIf
     EndForAll
EndWhile
```

However, the group creation can be achieved faster by using the *n*-tree. It is not necessary to explore the entire set C to find adjacent cells. Instead, it is possible to explore the tree.

To find the adjacent cells of a cell c taken at random in C and removed from it, start from the root cell and look for all child cells that could intersect or contain an area around the cell c. Do this recursively until you reach leaf cells. Only in this restricted set of cells, search for adjacent cells. Restart this procedure for each newly found cell until no more adjacent cell is found.

```
Algorithm GroupCells2( C ) returns G
C: set of cells
G: set of sets of cells
c, n: cell
g, h: set or cells
While not C.isEmpty do
     g <- new group
     h <- new group
     c <- C.removeRandom
     G.add(g)
     g.add(c)
     h.add( neighborsOf( tree.getRoot, c, C, area ) )
     While not h.isEmpty do
          ForAll cell d in h do
               If d.shareFaceOrEdgeOrPoint( g )
               Then
                    C.remove( d )
                    h.remove( d )
                    g.add( d )
                    h.add( neighborsOf( tree.getRoot, d,
C, area ) )
               Else
                    h.remove( d )
               EndIf
          EndForAll
     EndWhile
EndWhile

Method NeighborsOf( n, c, C, tree, area ) returns h
c, n: cell
C, h: set of cells
tree: n-tree
area: float
If not n.isLeaf then
     ForAll cell d in n.subCells do
          If n.containsOrOverlap( area, c ) then
               h.add( NeighborsOf( d, c, C, area ) )
          EndIf
     EndForAll
Else
     If C.contains( n ) then
          h.add( n )
     EndIf
EndIf
```

The difficulty in this method is that it is difficult to compute which cells are intersecting with the area around a leaf cell. A simple approximation can be to use a cube or square instead of a sphere or disk around the considered cell.

RESULTS

The results presented here are in two dimensions. This allows an easy representation of the levels. On each figure, a boid simulation is shown on the left and the space subdivision is shown on the right.

The boid simulation uses several species that repulse one another and form groups of high interaction. The space subdivision is in two dimensions, with a third dimension showing the depth in the quadtree. Levels close to the root are in darker and near the leaves are lighter. The cut threshold has been set to 5 levels, and levels higher than this have been colored in red, they show the organizations detected. The maximum number of bodies per cell is ten.

The Figure 6 show the boid simulation at different time steps.

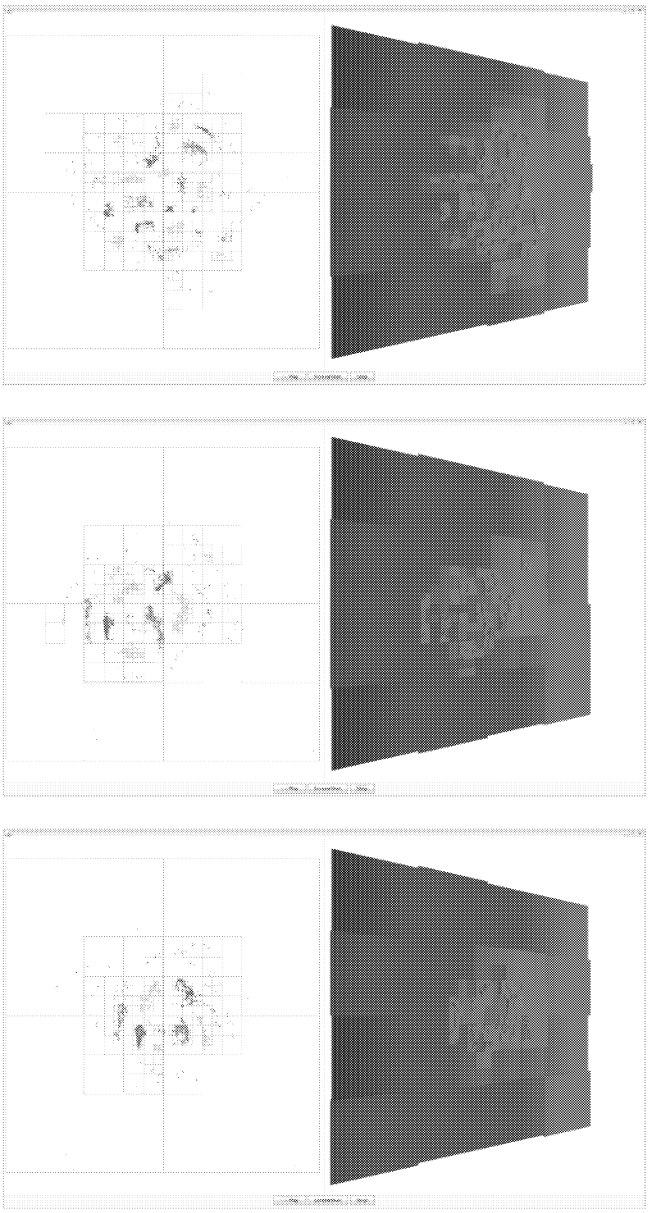


Figure 6 : Left, the boids in different colors. Right, the organizations detected, in red

CONCLUSION

We presented an organization detection method based on the reuse of the n-tree of fast multipole methods. Such methods are used in a wide range of problems. It offers a fast algorithm that uses the calculations of another one to speed its computation. The detected groups of bodies are close approximation of the real organizations.

REFERENCES

Beatson R. and Greengard L. *A short course on fast multipole methods*, http://math.nyu.edu/faculty/greengar/. Link is alive on 09/28/07.

Coulomb C.-A. 1785, *Premier Mémoire sur l'Électricité et le Magnétisme,* Paris

Dutot A., *Distribution dynamique adaptative à l'aide de mécanismes d'intelligence collective*, PhD thesis, Le Havre University

Frontier S., Pichot-Viale D.., *Écosystèmes, Dunod, 1998.*

Leonard A. 1980, « Vortex methods for flow simulation », *Journal of Computational Physics*, vol. 37, 1980, p. 289-335.

Newton I., 1686 *Philosophiae Naturalis Principia Mathematica*, London.

Reynolds, C. W., 1987. "Flocks, Herds, And Schools: A Distributed Behavioural Model". *Computer Graphics*. Vol. 21, No.4.

Simon H. 1996, *The Sciences of the Artificial (3rd Edition)* MIT Press

Tranouez Pierrick 2005, *Penicillo haere, nam scalas aufero*, PhD thesis, Le Havre University

Tranouez P., Bertelle C. and Olivier D. 2005, « Changing levels of description in a fluid flow simulation", EPNADS 2005

Design and Validation of a Distributed Computation Environment for Mobile Devices

Ian Oliver

Nokia Research Center
Itämerenkatu 11-13, Helsinki, Finland
ian.oliver@nokia.com

Abstract

Development of complex software systems is known to be difficult. The use of formal methods in such developments is advantageous but often suffers from difficulty in applying such methods and the visualisation of the results. We show here that the use of a light-weight formal specification language - Alloy - and the use of simulation traces similar to that of object-diagrams in UML can be effective in understanding the behaviour of such systems at analysis and design-times.

1 Introduction

The abundance of personal, mobile devices such as mobile phones or laptop computer with significant computational power, large memories - both temporary and persistent, multi-tasking operating system and with high-speed wireless communication facilities such as UMTS, WLAN and Bluetooth (see table 1) provides the potential for a sophisticated, distributed computation environment.

The Linda communication model defines a message passing method for distributed systems. This model has been utilised and augmented in various implementations and has now found favour as a mechanism in web service/semantic web/Web 2.0 architectures such as WSMX [19], JavaSpaces[1] and TSC [5].

We have an architecture based upon an augmented Linda system for the linking together of processes running upon various mobile devices though the notion of a 'knowledge space' (cf: [12]). The Linda model is augmented to provide a publish-subscribe mechanism with the notion of passing *facts* (items of *knowledge*) rather than messages. This provides us with an implicit communication method.

We chose to use as the development method for this particular system a pragmatic formal approach using the Alloy specification language. Alloy was chosen for the ease of use, its ability to specify dynamic systems and its suitability for producing execution traces and analyses which are more akin to how many engineers visualise a system. In addition the Alloy language has a closer semantic correspondence with defacto standard languages such as the ubiquitous UML [8, 2, 9]. In some previous projects have used languages such as B in isolation for the development of such systems with varying results [17] due to the strictness of the B-Method approach.

In this paper we describe our 'Smart Space' system and concepts through the use of a domain modelling approach and formalisation in Alloy along with various simulations of the operations of that system. We then make discussion on the validation of the demarcation or policy structure used in that system and following this describe how these models are integrated and simulated.

2 Smart Spaces

The *SmartSpace System* provides a method for concurrently running processes known as *Nodes* to interact with *SmartSpaces* which provide publish-subscribe mechanisms across a Linda-like space in which are contained *facts*. Figure 1 shows the structure of this described using the UML class diagram notation.

Applications are constructed from dynamic combinations of Nodes at design and run-times across multiple, heterogeneous devices. Nodes interact

[1] http://www.javaspaces.homestead.com/

Device	O/S	CPU	Memory	Storage	Communication
Nokia N95	Symbian	ARM11	160Mb	up to 4Gb	IR/BT/3G/WLAN
Nokia N800	Linux	ARM11	128Mb	Upton 4Gb	BT/WLAN
Nokia 6600	Symbian	ARM9	6Mb	32Mb+	BT/IR/3G
Apple iPhone	OSX	ARM9	shared	4-8Gb	WLAN/2G/BT
Laptop	Windows/Linux	x86	1Gb	100Gb+	WLAN/BT/IR

Table 1: Specification of Some Typical Mobile Devices

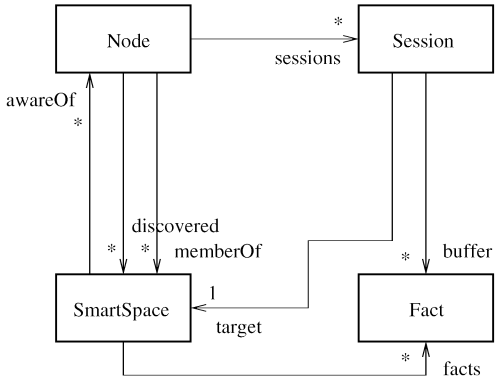


Figure 1: SmartSpace System Domain Model

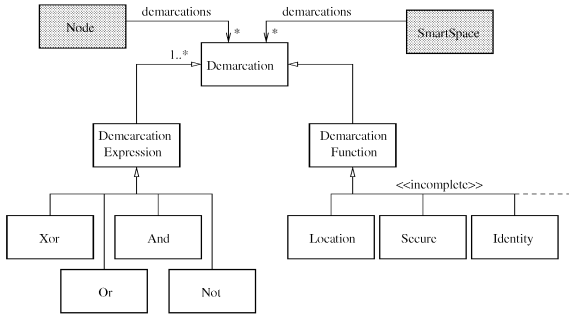


Figure 2: Demarcation Model in UML

with the SmartSpace through *Sessions* which provide facilities for the Node to add and remove facts and also subscribe and query for patterns of facts. Sessions can also provide additional pre/post-processing of the information as well as other functionalities relates to encryption and security. They also are used to hide the underlying communication mechanisms and technologies.

In order to interact with a SmartSpace various mechanisms to associate Nodes and SmartSpaces together are utilised. A Node may actively search for SmartSpaces through a discovery mechanism and once a SmartSpace has been found a Node may then request to explicitly join that SmartSpace. Similarly a SmartSpace may actively search for Nodes and subsequently invite those Nodes into the SmartSpace.

In addition to this there are various criteria which a Node has to satisfy in order to join a SmartSpace. A SmartSpace can be demarced by any number of criteria (fig.2) such as the positioning, acceptable security and identity credentials, and so on. In order for a join operation to be successful a Node must comply with these demarcation requirements.

The SmartSpace itself manages the facts input and requested for by the Nodes. These facts however are pieces of information which means that the normal rules of message passing as one normally sees in a Linda-type system do not apply - communication here is through implicit communication as a form of publish/subscribe. Typically in a Linda system the messages are either unordered (set based) or ordered (queue based) with duplication allowed and consumption of the messages explicit. We modify this to work with *information* which behaves in our current implementations according to the rules of the well-formed semantics.

3 Simulation and Validation of the Domain Model

In order to validate the requirements of the system we constructed a description using the Alloy specification language [13]. This description being based upon the model in figure 1. This model is then augmented by the addition of the notion of time and a method for ordering events through time as a trace; it is this ability to simulate particular traces that allows us to explore particular interesting scenarios.

The traces are constructed depending upon the needs of that simulation run. We do provide a description of an empty initial state which is used when we wish to simulate whether it is possible to reach a certain state, otherwise we let the Alloy tool choose any suitable starting state which admits the events we wish to run. In this case it is not guaran-

teed that the said starting state is reachable from an empty initial state though we can check for this. An example trace predicate that admits any starting state and one explicit event per time step is given below:

```
fact Traces {
   all e:Event |
      e.t' = e.t.next
      all x: Time - last |
         one e:Event | e.t = x }
```

The second modelling choice is to choose the amount of concurrency: we can choose at each step in time to simulate a single event or potentially multiple events occurring simultaneously; generally we choose the former for simplicity in reading the resulting traces. Furthermore when expressing the pre and post-conditions of the events we are free to decide on the extent of the frame-rule [4] for those events. In a sequential, atomic semantics as seen in languages such as B/EventB [1] the frame-rule is explicit and states that 'nothing else happens' however in the case of a distributed system with many points of concurrency this can prove too restrictive and mean that some interesting properties and situations may be lost; the disadvantage of this being that with concurrency it can be difficult to see what is supposed to happen with the parts of the system you are specifically interested in. Furthermore restricting the concurrency, especially in the distributed system, at specification-time has deep architectural implications later on in the development with regards to synchronisation protocols across the system.

3.1 Behaviour Simulation

The static structure of figure 1 was simply translated into Alloy as a signatures and relationships between signatures. The events upon the model were expressed as first class signatures in their own right leading to a structure based upon the responsibility and scope of the event, this is shown in figure 3. All events are expressed across two points in time: **t** and **t'**, from this are derived specialisations with additional parameters and constraints or frame-rules if necessary as shown by the NSessFact_Event signature:

```
abstract sig Event {t,t':Time}

abstract sig NSessFact_Event extends Event{
n:Node,
se:Session,
f: one Fact }
   {    only_NseF[t,t',n,se]  }

pred only_NseF (  t,t':Time, n:Node, se:Session){
 let others = Node - n {
  others <: discovered.t' =
   others <: discovered.t
  aware_of.t' :> others =
   aware_of.t :> others
  others <: memberOf.t' =
   others <: memberOf.t  }
}
```

The two events used in this example are given below. The first states that when a fact is sent by a node to a space via a given session that the session is 'owned' by that node, the fact is not a duplicate of a fact already in the buffers and the fact does not already exist in the system a whole. These facts appear strange but ensure that this operation selects a new fact that has not already been used in the system to this point. The result of this event is to add that fact to the buffer of the session.

```
sig SendFact extends NSessFact_Event{}
{
   se in n.sessions.t
   all sessions:Session |
      f not in sessions.buffer.t
   all s : SmartSpace |
      f not in s.facts.t
   se.buffer.t' = se.buffer.t'  + f
}
```

The following event checks that the given buffer is not empty in the before-state and in the after-state ensures that the buffer of that session is empty and that the facts formerly held in the buffer are transferred to the space.

```
abstract sig SchedulerSession_Event extends Event{
  se:  Session,
     ss: SmartSpace  }{}

sig SchedulerTakeFacts extends
   SchedulerSession_Event{}
{
   #se.buffer.t > 0
   ss.facts.t' = ss.facts.t + se.buffer.t
   no se.buffer.t'
}
```

To drive the simulation one instructs Alloy to 'run' for a number of time steps without any preference to which events should occur; this provides an idea of how the system might behave. Typically we look for particular scenarios such as when a Node sends

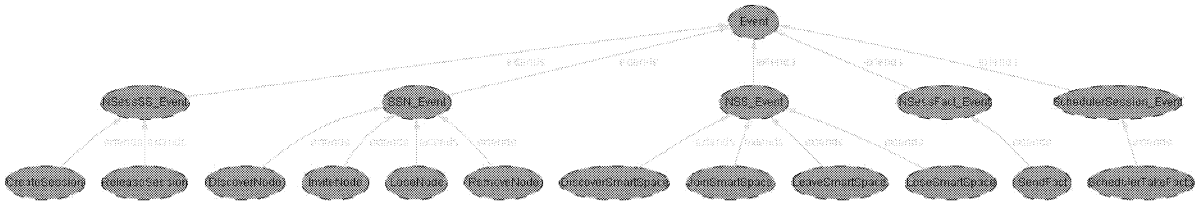


Figure 3: Event Structure

a fact via some Session and the SmartSpace scheduler collects the waiting facts from that Session's buffer (see event declarations above) as shown in the following example.

```
run { some e:SendFact, f:SchedulerTakeFacts,
    s:Session | e.t' = f.t and e.se = f.se
       and #s.buffer.(e.t)>2
       and s.buffer.(e.t) in s.buffer.(f.t)
 } for 5 but 3 Time,2 Session
```

We also restrict this simulation run to where the session buffer at the start contains more than two facts already and that nothing else changes the state of the buffer in the meantime - at least nothing is removed and perform this for a maximum of 5 objects of each type, 3 points in time (number of events + 1 at least) and 2 sessions (for convenience). From this we obtain three models[2] one for each point in time: figures 4,5 and 6.

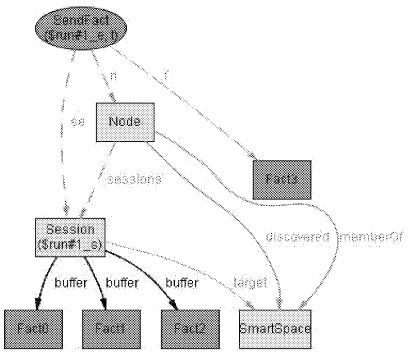


Figure 4: Simulation (1) - Time 0

It is seen from this simulation that all the facts are indeed transferred across to the SmartSpace and nothing is left in the Node's buffer. If no trace is found then Alloy generates a counter-example which then details where the trace fails.

[2]Actually only one model is produced but we project over the Time to obtain graphs for each of the time points individually

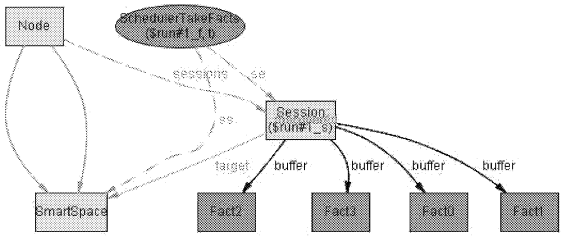


Figure 5: Simulation (1) - Time 1

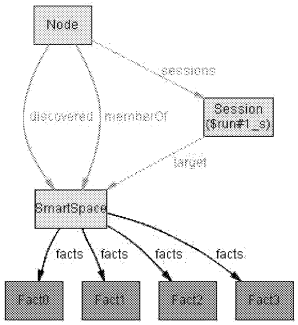


Figure 6: Simulation (1) - Time 2

Note that Alloy only checks a finite state space and is used in according with the principles of the 'small scope hypothesis' [3] which states that most errors can be found within small examples and that it is more effectively to perform systematic validation with smaller scopes than to work with fewer tests over larger scopes. Our experience has been that given the size of events and structures that these events are operating over simulation each one in small cases provides a much larger body of easily comprehensible information rather than having to understand the results of one much larger and longer simulation.

We can explore traces related to concurrency by weakening the trace predicate to admit more than one event per time step but running the same query - an example simulation then produces the models

as in figures 7,8 and 9.

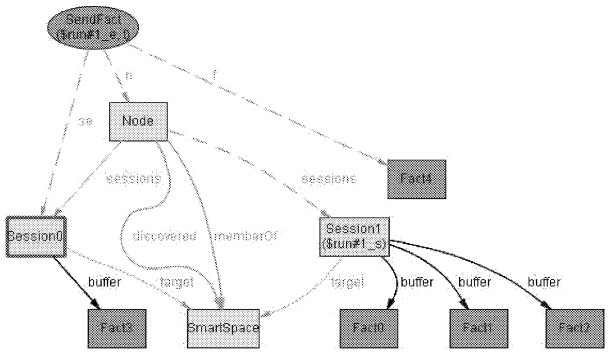


Figure 7: Simulation (2) - Time 0

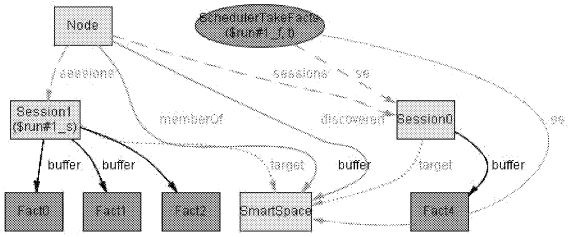


Figure 8: Simulation (2) - Time 1

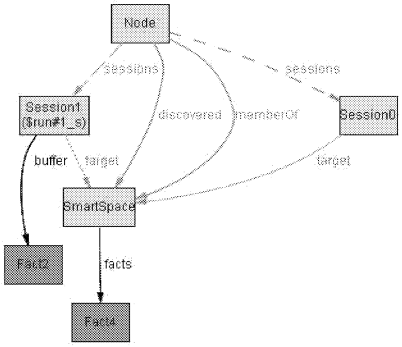


Figure 9: Simulation (2) - Time 2

In this case we have a more complex example generated with two sessions both of which have populated buffers. It is interesting to node the during the course of sending Fact4 to the buffer of Session0, Fact3 is removed. This *may* be due to an error in the specification or a concurrent action removing the fact from the buffer. In this case we are confident that another event is occurring. Note that our declarations of the event only state what must be true before-hand and what must be true afterwards, we state nothing about other changes and the atomicity or transactional properties of the event and in this case the model in figure 8 *is correct* with respect to this. We can choose to restrict the concurrency by explicitly 'locking' the buffer during this operation through expression of a frame-rule for this particular event and then rerun the simulation to check the outcome - the results of this simulation would be that the situation seen above would not occur as we would have enforced the atomicity of the event. As we are constructing a distributed system, enforcing atomicity over multiple nodes implies signalling and synchronisation protocols which may not be possible and/or desirable to write.

A similar but even more dramatic situation occurs during the course of the second event resulting in figure 9 where as before the fact is transferred from the buffer to the SmartSpace but also facts are removed from the buffer of the unrelated Session1. Again this is due to the concurrency/frame-rule issues which may again be desirable to keep weak thus improving the concurrent properties of the system as a whole. To adequately investigate this further simulation runs under differing conditions would of course need to be run.

This is a good example of the small scale hypothesis where it is possible to see the effects of the concurrency in a restricted domain where it is easily decidable whether the concurrency and its effects are desirable. This is something that is extremely difficult or even impossible to ascertain when presented with much larger models where the difference is difficult to distinguish.

4 Demarcation Model

The demarcation model in figure 2 describes the structure policy mechanism by which Nodes are validated before admission to a given SmartSpace. Translation of this UML model into Alloy is straightforward and Alloy presents us with the ability to generate particular instantiations of this model; an simple example of which is shown in figure 10.

Through the visualisation of this model we can easily see what particular combinations are possible. By further constraining the model we can disallow certain combinations, for example, locations and identities may not appear in the same demarcation. Obviously as further subtypes of the demarcation functions (specialisation of location etc) are added then these models and their constraints become more complex. If we over-constrain a model Alloy provided us with a counter-example demon-

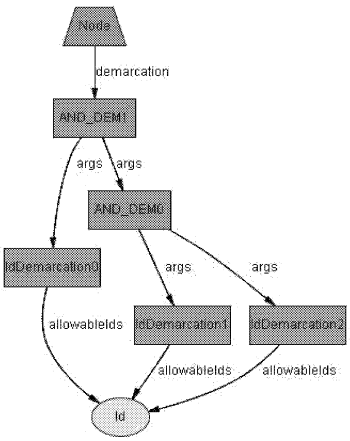


Figure 10: Demarcation Model Instantiation Example

strating that particular case.

5 Integration of Models

Our development has progressed with two separate models of differing parts of the system, though it can be seen from figure 2 that there is a relationship between these models. Obviously some integration must be performed and the integration itself must be validated. The integration also implies a development step in the model and in this case the joining mechanism between a Node and Space must be updated to admit this policy mechanism.

The first step that must take place is changing the join operation by adding a precondition to check the correspondence between the Node's credentials and the demarcation properties of the Space. A join may only occur if the node is not a member of that particular space and the respective policies match. We also ensure that the node has also discovered the space before-hand and if all of the above are true then the node becomes a member of that space.

```
sig JoinSmartSpace extends NSS_Event{} {
    policyCheck[n,ss]
    ss not in n.memberOf.t
    ss in n.discovered.t
    n.memberOf.t' = n.memberOf.t + ss
}

policyCheck(n:Node,ss:SmartSpace) {
    n.demarcation in ss.demarcation
}
```

For the purposes of simulation and ease of specification we do make a simplification at this stage and ignore the various logical operations that could occur over the demarcation policy - hence the trivially simple policy check predicate. As the purpose of the simulation is to visualise potential situations and understand the join operation in the presence of policy information we consider this to be acceptable[3]. Indeed experience has shown that overloading the specification with precise details at this stage - especially when the simulation is to be presented to both engineers and non-engineers alike - just complicates the discussion about the system as a whole rather than concentrating on the essentials of the operation under investigation.

Alloy in the successful case will generate two points in time for the before and after states of the successful join operation as shown in figures 11 and 12. If we continue the time steps after this event we notice that the system behaves as previously. A more thorough check would involve simulating the cases previously without the demarcation criteria and comparing them more deeply.

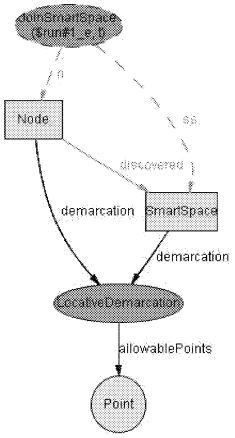


Figure 11: Join Event with Demarcation - Time 0

We can also force the system to check if join is possible if the demarcations do not match. In this case as the specification suggests and the simulation run demonstrates no successful instantiation of the model in this circumstance is possible.

Of course successful integration of the models also means careful modelling at similar abstraction levels though this is an area where Alloy helps because of the simplicity and style of specification. In the case here the work required to 'bind' the two models together involved understanding that the con-

[3]for the moment - again simple (simplified) and small examples

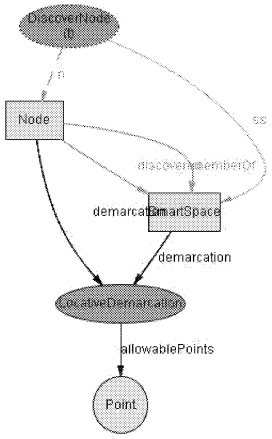


Figure 12: Join Event with Demarcation - Time 1

cept of Node in both was the same, the addition off a predicate to strengthen the join event and also to add a link to a demarcation structure from the Space.

6 Related Work

The use of formal methods in software engineering is not new - and Alloy has compelling examples of industrial usage[4] - however the use of more pragmatic methods which do not rely on an absolute strict method of constructing the models coupled with the ability to 'execute' or simulate models is less frequent.

As a comparison work made as part of the EU Rodin project [6] utilising models written in the B/EventB languages and coupled with the CSP [11] language to drive those models using the tool ProB [15] is very much in the same manner as explained here.

One major difference is that the use of CSP means that we are driving the model in a very specific way: we are stating that we wish to see if the model admits the trace as specified by the CSP rather than asking the model if such a trace is possible at all rather than from a particular given state. This means we can not explore individual states for their correctness - this being proven by the theorem prover before-hand - but checking now the overall progress of states occurs and whether that particular trace is admissible.

[4] http://alloy.mit.edu/publications.php

7 Conclusions and Future Work

Unlike existing solutions from formal methods tools in the software engineering domain, Alloy allows simulations and the visualisation of those simulations of the behaviour of a system in a manner that is more compatible with the industrial environment . That is, systems work by single stepping through the actions over a period of time. This coupled with Alloy's much freer form of specification - one where the underlying action system is not constrained by a preconceived action semantics with additional constraints such as frame-rules etc makes the ability to explore the system even in the presence of concurrency and additionally makes the simulation even in potentially distributed environments more tractable.

Of course, other formal methods and languages have supporting tools such as Z/EVES [7], ZANS [14] etc, the most relevant of these is the aforementioned ProB tool. B/EventB however suffers from the fact that there is a strict development method and particular semantics attached that makes it unsuitable or at least more difficult and unnatural in our development environment for producing the kinds of simulations shown here.

One particular advantage of Alloy's style of specification and simulation is that it is very similar to the ubiquitous UML language extensively used within industry. The traces that Alloy produces are almost identical to the use of object diagrams that are employed in a similar manner [16]. UML however does not have any suitable simulation tools that scale to industrial use well, though a mention should be made of the USE tool [10] which does support in part the OCL which is used in conjunction with the UML for formally expressing actions.

Regarding the overall effect of such a simulation the development of our software, as described in [18], any analysis of the system made early in the development flow has advantages. In this case Alloy affords us the ability to simulate in a more explorative way than competing formal tools. For the engineer, the ability to see the effects of the operations allows much earlier feedback and subsequent modifications to the design coupled with a greater understanding of what is trying to be achieved.

8 Acknowledgments

To Daniel Jackson for much assistance and help with the Alloy language - especially the trace predicates - and Jukka Honkola and Jürgen Ziegler at Nokia Research with the SmartSpace project itself.

References

[1] J-R Abrial. *The B-Book - Assigning programs to Meanings*. Cambridge University Press, 1995. 0-521-49619-5.

[2] Kyriakos Anastasakis, Behzad Bordbar, Geri Georg, and Indrakshi Ray. *UML2Alloy: A Challenging Model Transformation*. In *ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems (MoDELS)*. 2007.

[3] A. Andoni, D. Daniliuc, S. Khurshid, and D. Marinov. *Evaluating the small scope hypothesis*, 2002.

[4] Alex Borgida, John Mylopoulos, and Raymond Reiter. *On the Frame Problem in Procedure Specifications*. IEEE Transactions on Software Engineering, 21(10):785–797, October 1995.

[5] Christoph Bussler. *A Minimal Triple Space Computing Architecture*. In *Proceedings of the 2nd WSMO Implementation Workshop (WIW 2005), Innsbruck, Austria*. June 6-7 2005.

[6] Joey Coleman, Cliff Jones, Ian Oliver, and Elena Troubitsyna Alexander Romanovsky. *RODIN (Rigorous Open Development Environment for Complex Systems)*. In *Fifth European Dependable Computing Conference EDCC-5, Budapest*. 2005.

[7] Dan Craigen, Sentot Kromodimoeljo, Irwin Meisels, Bill Pase, and Mark Saaltink. *EVES: An Overview*. In *VDM'91 - Formal Software Development Methods - Lecture Notes in Computer Science 551*. Springer-Verlag, 1991. Noordwijkerhout.

[8] Michelle Crane and Juergen Dingel. *Runtime Conformance Checking of Objects Using Alloy*. In *Third Workshop on Runtime Verification (RV'03). Boulder, Colorado*. July 2003.

[9] G. Georg, J. Bieman, and R. France. *Using Alloy and UML/OCL to specify run-time configuration management: a case study*. In *Proc. Workshop on Practical UML-Based Rigorous Development Methods — Countering or Integrating the eXtremists*. October 2001.

[10] Martin Gogolla and Mark Richters. *Development of UML Descriptions with USE.*. In Hassan Shafazand and A. Min Tjoa, editors, *EurAsia-ICT*, volume 2510 of *Lecture Notes in Computer Science*, pages 228–238. Springer, 2002. ISBN 3-540-00028-3.

[11] C A R Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[12] Kit-Ying Hui, Stuart Chalmers, Peter Gray, and Alun Preece. *Experience in Using RDF in Agent-mediated Knowledge Architectures*, pages 177–192. Springer, Germany, 03 2004.

[13] Daniel Jackson. *Software Abstractions: Logic, Language, and Analysis*. MIT Press, April 2006. 0-262-10114-9.

[14] Xiaoping Jia. *A Tutorial of ZANS - A Z Animation System*. DePaul University, Chicago, USA, May 1995. available from ise.cs.depaul.edu/pub/ZANS by anonymous FTP.

[15] Michael Leuschel and Michael Butler. *ProB: A Model Checker for B*. In *FME 2003: Formal Methods*, pages 855–874. 2003. ISBN 3-540-40828-2.

[16] Ian Oliver. *Simulation of Software Behaviour using Animation.*. In Krzysztof Amborski and Hermann Meuth, editors, *Proceedings of European Simulation Multiconference*, pages 278–283. SCS Europe, 2002. ISBN 90-77039-07-4.

[17] Ian Oliver. *Experiences in Using B and UML in Industrial Development*. In Jacques Julliand and Olga Kouchnarenko, editors, *B*, volume 4355 of *Lecture Notes in Computer Science*, pages 248–251. Springer, 2007. ISBN 3-540-68760-2.

[18] J. Christopher Westland. *The cost of errors in software development: evidence from industry*. The Journal of Systems and Software, 62(1):1–10, May 2002.

[19] M. Zaremba and M. Moran. *Enabling Execution of Semantic Web Services: WSMX Core Platform*. In *Proceedings of the 1st WSMO Implementation Workshop (WIW2004) Frankfurt, Germany*. 2004.

TRANSPORT AND TRAFFIC FLOW

A SIMULATION APPROACH TO THE ANALYSIS OF INTERMODAL FREIGHT TRANSPORT NETWORKS

An Caris
Gerrit K. Janssens
Transportation Research Institute
Hasselt University - campus Diepenbeek
Wetenschapspark 5 - bus 6, 3590 Diepenbeek, Belgium
e-mail: {an.caris,gerrit.janssens}@uhasselt.be

Cathy Macharis
Department MOSI - Transport and Logistics
Vrije Universiteit Brussel - Managementschool Solvay
Pleinlaan 2
1050 Brussel, Belgium
e-mail: Cathy.Macharis@vub.ac.be

KEYWORDS
Discrete event simulation, methodology, intermodal transport, inland navigation, network analysis

ABSTRACT

In regions with an extensive waterway network intermodal transport including inland navigation is a good alternative for unimodal road transport. A discrete event simulation methodology is proposed to analyze policy measures with the intention of stimulating intermodal barge transport. The intermodal hinterland network of the port of Antwerp serves as the real-world application in this study. Various aspects in the modelling process are discussed and a first potential policy for analysis is proposed.

INTRODUCTION

In this paper a discrete event simulation methodology is developed to capture and analyze the interactions in intermodal freight transport networks. Macharis and Bontekoning (2004) define intermodal transport as the combination of at least two modes of transport in a single transport chain, without a change of container for the goods, with most of the route travelled by rail, inland waterway or ocean-going vessel and with the shortest possible initial and final journeys by road. Intermodal transport may include various transport modes. Our objective is to analyze the performance of intermodal networks including inland navigation and road transport. In regions with an extensive waterway network, such as Western Europe, intermodal transport including inland navigation is a good alternative for unimodal road transport.
Intermodal planning problems are more complex due to the inclusion of multiple transport modes, multiple decision makers and multiple types of load units. Because of this increased complexity and the required level of detail, discrete event simulation is the appropriate tool of analysis. A simulation model will be created to support decisions in intermodal transport at the strategic level. Simulation models have been widely used to optimize the design of intermodal terminals. For example, Rizzoli et al. (2002) present a simulation tool for the combined rail/road transport in intermodal terminals. Parola and Sciomachen (2005) describe a strategic discrete event simulation model to analyze the impact of a possible future growth in sea traffic on land infrastructure in the north-western Italian port system. We develop a simulation model that covers the hinterland waterway network of a major port in Western-Europe in order to analyze effects of future policy measures for intermodal container transport. A first policy related to the consolidation network is presented in this paper. Various studies discuss consolidation strategies for intermodal transport by rail, see for example Janic et al. (1999) and Newman and Yano (2000). Konings (2005) presents and evaluates a consolidation strategy for intermodal transport by barge, based on a marginal cost model. In our simulation model the operations of the inland navigation network are modelled in detail. This enables us to examine ex-ante what the effects of a certain consolidation strategy will be and to take into account interaction effects in container flows. In the future, the model will also be used to analyze other policies related to intermodal barge transport. This paper presents current research which is still in progress. The simulation model will be part of a larger decision support system for intermodal transport policy making which is not yet discussed here.

MODELLING METHODOLOGY

In this section the methodology to model the hinterland waterway network is described. First, the main characteristics of the intermodal network under investigation are given. Next, the conceptualization of the network is developed. The inputs and outputs of the simulation model are discussed. In the final subsection special attention is given to the functioning of the locks.

Intermodal Transport Network

The intermodal hinterland network of the port of Antwerp serves as the real-world application in our

study. Three regions of origin can be identified in the network. The first group of container terminals is situated along the Albert Canal towards the eastern part of Belgium. A second region of origin is located in the central part of the country, connected to the port of Antwerp by the Brussels - Scheldt Sea Canal. The third group of intermodal container flows originates in the basin of the Upper Scheldt and the river Leie. All intermodal container terminals organize shuttle services either to the port of Antwerp or to the ports of Rotterdam and Amsterdam. Two clusters of sea terminals can be identified in the port area of Antwerp. Until recently the main center of activity was situated on the right river bank. With the construction of a new dock (Deurganckdok) in the port of Antwerp, a second cluster of sea terminals emerged on the left river bank. Barges sail through the Scheldt-Rhine connection to Rotterdam and Amsterdam. A last destination is the port of Zeebrugge, which can be reached via Antwerp and short sea shipping on the river Scheldt. Table 1 summarizes all origins and destinations of shuttle services. Shuttle services transport containers from inland terminals to sea terminals in the port area and carry containers from sea terminals to inland destinations in a round trip.

Table 1: Origins and Destinations

Origins	Destinations
Albert Canal	Antwerp: right river bank
Brussels-Scheldt Sea Canal	Antwerp: left river bank
Upper Scheldt and Leie	Rotterdam
	Amsterdam
	Zeebrugge

Conceptual Model

Three major components can be identified in the intermodal hinterland network, as depicted in figure 1. The first component in the intermodal freight transport network is the inland waterway network. The inland waterway network is made up of terminals, waterway connections and container flows. Entities are defined as barges which originate from the different inland terminals and carry containers in round trips to the various ports. A second component is the port area of Antwerp. Barges may visit sea terminals at the left river bank and right river bank in the same round trip, go to Rotterdam or Amsterdam via the Scheldt-Rhine connection or sail to Zeebrugge via the Scheldt estuary. On the right and left river bank, barges queue for handling at the sea terminals. Barges moor as soon as enough quay length is available. The handling time at the sea terminal depends on the number of containers that need to be unloaded from or loaded into the inland vessel. In the inland waterway network as well as in the port area multiple locks are present. Therefore, the lock planning constitutes a third major component, which will be discussed hereafter.

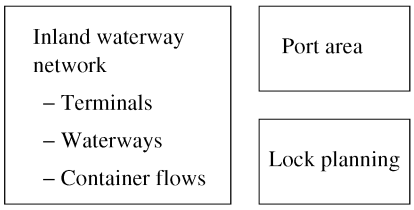


Figure 1: Components

The objective of the model is to simulate possible policy measures for intermodal barge transport. Consequences and implications can be estimated before implementation of the policy measure. Therefore, various conceptual models may be necessary to analyze the implications of proposed policies. The conceptual model of the current container flow is depicted in figure 2. At present all barges enter the port area and visit one or multiple sea terminals.

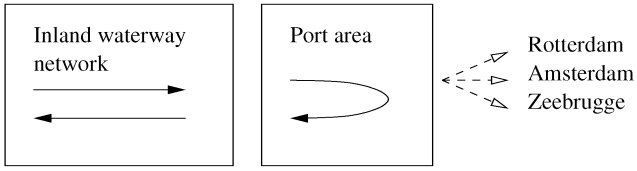


Figure 2: Conceptual Model Current Situation

Assumptions

A number of assumptions are made to translate the actual intermodal network into a discrete event simulation model. The emphasis lays on inland waterway transport. Rail connections in the hinterland network are not taken into account. The model further assumes a homogeneous container type and equal handling time for each container. All main waterway connections between inland terminals and the port area are incorporated in the simulation model. Small waterways without inland terminals are not included in the simulation model of the current situation. Sailing times are assumed to be stochastic and follow a probability distribution. The average sailing time varies with the type of barge. A probability distribution is also used to model the stochastic lockage time.

Data Requirements

All intermodal terminals in the inland waterway network are asked for information to identify the container flows. Real data on shuttle services is used as an input

for the simulation model, constructed in the simulation software Arena. For each shuttle service the following information is required: which type of barge is used, which destinations are visited and what is the average number of import and export containers for each destination. Container transport interacts with other freight flows. Therefore, the flow of non-containerized goods on the inland waterway network is introduced as an input in the simulation model. These flows affect the waiting times at locks. Information is also necessary on the network connections. The waterway administrators provided information on the number of locks on each waterway, distances between locks, average lockage times, number of lock chambers and size of the chambers. In the port area of Antwerp three clusters of locks connect the inner port area with the sea side. Data is required on the choice of locks when sailing in the port area. The average quay length available for handling inland navigation at sea terminals gives an indication of the service capacity in the port area of Antwerp. The port authority is asked for the average mooring time and time for loading and unloading in order to model the service times of inland container barges in the port area. Finally, an enquiry is made into the turnaround times of vessels and average waiting times at locks in order to verify and validate the model.

Outputs

Table 2 gives an overview of performance measures which are generated by the simulation model. The turnaround time of shuttles is defined as the total time necessary for a barge to sail from an inland container terminal to the port area, visit all sea terminals and return to the inland terminal. The turnaround time depends on the waiting times at locks and in the port area. The outputs measured at locks are the percentage of barges that have to wait, the number of barges that have to queue and the waiting time of barges in the queue. In the port area the waiting time before handling is measured, as well as the number of vessels queueing for service. A final group of performance measures concerns the capacity utilization. In the port area this is expressed as the average percentage of quay length occupied. In the hinterland network the average and maximum number of barges on each network connection is recorded.

Locks

The operations of locks strongly affect waiting times of barges for lockage. A number of decision rules are defined to make the operations of the locks in the simulation model reasonably realistic. A first decision rule relates to the size of a barge. Barges are assigned to a lock chamber only if its size is within the allowed dimensions. Secondly, barges are assigned to the smallest lock chamber that is open. This decision rule focuses on a rapid lockage process of barges. Smaller lock chambers have a shorter lockage time. On the other hand, a more intensive use of larger lock chambers may reduce waiting times because more barges can be serviced simultaneously. A third decision rule is applied when no lock chamber is open in the sailing direction of the barge. In this situation the barge is assigned to the lock chamber which is the first available. A final decision rule concerns the closing of lock chambers. A lock chamber is closed when there is not enough remaining space for the next barge in queue or when no additional barges arrive within a predefined number of time units. From interviews with waterway administrators it appears that the operations of locks are entrusted to a lockkeeper, without fixed rules. Future research could introduce more complex decision rules in the simulation model.

Table 2: Performance Measures

Shuttles	turnaround time
Locks	total number waiting (%) number waiting in queue waiting time in queue
Port area	waiting time in queue number waiting in queue
Capacity utilization	quay length network connections

ANALYSIS OF CONTAINER FLOWS

The simulation model will be used to analyze policy measures that impact container flows in the intermodal hinterland network and the port area of Antwerp. It will further allow to detect future bottlenecks in the infrastructure of the network. To this end various scenarios of increasing transport volumes can be modelled.

One potential policy might be the introduction of an intermodal barge hub in the port of Antwerp, from which load is distributed to the different sea terminals, as proposed by Konings (2005). The author proposes to split existing barge services into a trunk-line operation in the hinterland and collection/distribution operations in the seaport. This leads to the conceptual model depicted in figure 3. By doing so inland barges do not have to call at multiple sea terminals. They only visit the intermodal barge hub. This leads to a reduction in turnaround time of vessels serving the hinterland. In the collection/distribution network containers with the same origin or destination can be bundled. This enables a more efficient and prompt handling of barges at sea terminals. The simulation model will allow to demonstrate to what extent the waiting times in the port area and the turnaround time of inland barges can be reduced.

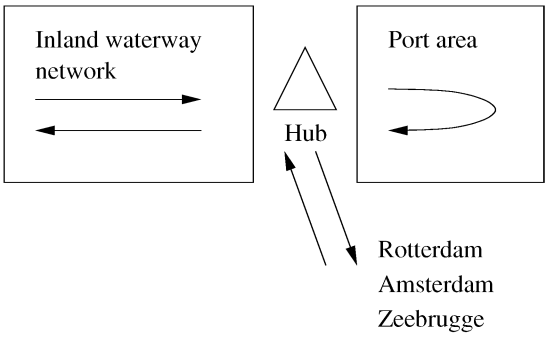


Figure 3: Conceptual Model Future Situation

CONCLUSIONS AND FUTURE WORK

The modelling process is presented of a discrete event simulation model for an intermodal freight transport network. The model will be used to make a quantitative ex-ante analysis of policy measures to stimulate intermodal barge transport. In this study the main focus is on the inland waterway network. Potential improvements include the introduction of more complex decision rules for the operations of locks. A submodel could also be introduced to integrate the intermodal terminal planning into the simulation model.

Acknowledgement: This research is supported by a grant from the Belgian Science Policy in the research programme "Science for a Sustainable Development", under contract number SD/TM/08A.

REFERENCES

M. Janic, A. Reggiani, and P. Nijkamp. Sustainability of the European freight transport system: evaluation of innovative bundling networks. *Transportation Planning and Technology*, 23:129–156, 1999.

R. Konings. Smart collection and distribution of containers on barge in the port of Rotterdam. *Proceedings of the Nectar Conference 2005*, Las Palmas, Gran Canaria, 2005.

C. Macharis and Y. M. Bontekoning. Opportunities for OR in intermodal freight transport research: A review. *European Journal of Operational Research*, 153: 400–416, 2004.

A. M. Newman and C. A. Yano. Scheduling direct and indirect trains and containers in an intermodal setting. *Transportation Science*, 34(3):256–270, 2000.

F. Parola and A. Sciomachen. Intermodal container flows in a port system network: Analysis of possible growths via simulation models. *International Journal of Production Economics*, 97:75–88, 2005.

A. E. Rizzoli, N. Fornara, and L. M. Gambardella. A simulation tool for combined rail/road transport in intermodal terminals. *Mathematics and Computers in Simulation*, 59:57–71, 2002.

BIOGRAPHY

AN CARIS graduated as Master of Business Economics with a major in Operations Management and Logistics at the Limburg University Centre (LUC), Belgium, in 2003. After one year of practical experience in inventory management at Reynaers Aluminium, she started as a teaching assistant in econometrics and operations research at the Hasselt University (UHasselt), Belgium. In addition she is preparing a Ph.D. in Applied Economic Sciences. She is a member of the research group Data Analysis and Modelling and the Transportation Research Institute (IMOB) of the Hasselt University. She takes a research interest in modelling intermodal freight transport networks with a focus on inland navigation.

GERRIT K. JANSSENS received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University within the Faculty of Business Administration. He holds the CPIM certificate of the American Production and Inventory Control Society (APICS). During the last fifteen years he has repeatedly been visiting faculty member of universities in Thailand, Vietnam, Philippines, Cambodia and Zimbabwe. His research interests include the development and application of operations research models in production and distribution logistics.

CATHY MACHARIS received her Master of Business Economics and Ph.D. degrees from the Solvay Business School, Free University of Brussels (VUB), Belgium. Currently she is professor at the Free University of Brussels (VUB), Faculty of Economics, Social and Political Sciences and Solvay Business School. She teaches courses in operations and logistics management, transport and business economics. Her research focuses on establishing linkages between advanced operations research methodologies and practical logistics challenges faced by senior management in firms and public agencies. She has been involved in several national and European research projects dealing with topics such as location selection, intermodal transport optimization, road safety enhancement through telematics applications, public transport and inland navigation.

A NONLINEAR TRANSPORTATION PROBLEM WITH ONE HUNDRED VARIABLES

William Conley
Department of Business Administration
University of Wisconsin at Green Bay
Green Bay, WI 54311-7001
U.S.A.
E-mail: conleyw@uwgb.edu

KEYWORDS

Transportation, nonlinear programming, optimization.

ABSTRACT

The classical mathematical transportation problem is one that attempts to deliver a fixed number of units of a product from the supplier's various warehouses (or storage facilities) to a number of customers at minimal cost or maximum profit, meeting all of the customers individual requirements. The classic transportation problem assumes different constant unit costs for each shipping route (from each different warehouse to each different customer). Then the problem is usually solved with some linear programming simplex computer package or one of the specialized versions of it designed specifically for the classic transportation problem.

However in the practical global business world, the unit shipping costs from each warehouse to each customer may vary greatly with the amount shipped. Therefore presented here will be a ten warehouse to ten different customer locations shipping problem where the unit shipping costs vary with the amount of product shipped. Therefore linear programming is replaced with the computer simulation based solution technique multi stage Monte Carlo optimization (MSMCO) to solve this specific problem presented here

THE HUNDRED VARIABLE PROBLEM.

A European company is shipping its product in bulk from many warehouses to many customers. However because the different unit shipping costs are not constant (they vary with the amount shipped from each location to each destination) the company decides to use the MSMCO simulation technique to solve its cost control problem.

The MSMCO Simulation approach makes repeated "random" samples of solutions in the feasible solution space, always storing the best answer so far and repositioning and further focusing the search in those areas as the MSMCO algorithm proceeds to a useable approximate solution.

Specifically in their example 10,500,000 units are to be shipped from ten warehouses to ten different customers. The shipping costs for this had been around five million euros in the past, so the goal this time (taking the discounts for shipping in quantity into account) is 3,500,000 euro dollars total shipping costs. The customer demands are 600,000 through 1,500,000 units in varying amounts, with different warehouse capacity amounts given below with the cost equation which has been developed to reflect the unit discounts for shipping in larger quantities. Therefore we seek to solve the following: 100 variable 21 equation nonlinear system of equations.

$$\sum_{i=1}^{10} \text{xij} = 500{,}000 + 100{,}000\ \text{j for j} = 1,2,3,... \quad 10\ \text{warehouse capacities} \qquad (1)$$

and

$$\sum_{\text{j}=1}^{10} \text{xij} = 1{,}600{,}000 - 100{,}000\,\text{i for i} = 1,2,3... \quad 10\ \text{customer demands} \qquad (2)$$

$$C = \sum_{i=1}^{10}\sum_{\text{j}=1}^{10} (2.80 + .05(\text{i} + \text{j}))x\,\text{ij} \exp(.5 + .025(\text{i} + \text{j})) = 3{,}500{,}000 \text{ where } x\,\text{ij} \text{ is the amount shipped from warehouse i to customer j.} \qquad (3)$$

by minimizing the sum of the absolute values of the difference between the left and right hand side of each equation. A thirty stage multi stage Monte Carlo optimization (MSMCO) simulation is attempted drawing 30,000 feasible solutions at each

stage and repositioning the search centered around the best answers so far. The search region is also narrowed appropriately after each stage. The idea is to funnel across one hundred dimensional space to an exact solution or a near approximation to the 21x100 system.

The thirty stage errors follow (in table 1) with a high of 104,132,112 (not good) in stage one down to fairly acceptable total errors of less than 1 in stage 25 through 30 (stage 30 error .125). Also right hand side constants of the 21 equations are printed (in table 2) along with the 100 x ij amounts to be shipped from warehouse i to customer j(in table 3) to achieve a total cost of 3,500,000 euros plus or minus one eighth of a Euro.

Table 1: Stage Errors of Printout

1	104132112.
2	71195952.
3	34194648.
4	7640255.
5	2001702.125
6	810924.25
7	347320.94
8	165024.16
9	49086.45
10	24321.37
11	13808.55
12	4837.19
13	3419.47
14	1672.19
15	867.77
16	423.06
17	202.20
18	108.36
19	61.96
20	26.28
21	11.41
22	6.02
23	3.48
24	1.31
25	.68
26	.27
27	.14
28	.125
29	.125
30	.125

Table 2: Equation Constants

Equation Number	Right Hand Sides
1	600000
2	700000
3	800000
4	900000
5	1000000
6	1100000
7	1200000
8	1300000
9	1400000
10	1500000
11	1500000
12	1400000
13	1300000
14	1200000
15	1100000
16	1000000
17	900000
18	800000
19	700000
20	600000
21	3500000

Table 3: Amounts Shipped from warehouses to customers.

Amounts i	j	X ij
1	1	46232.0117
1	2	55972.1641
1	3	3640.5823
1	4	18295.8965
1	5	55192.9805
1	6	55705.6016
1	7	53680.6836
1	8	98165.4453
1	9	348290.8750
1	10	764823.7500

Amounts i	j	X ij
2	1	5228.3330
2	2	109493.1875
2	3	14887.6875
2	4	21579.9316
2	5	46313.9570
2	6	45473.6328
2	7	109871.7891
2	8	340243.1875
2	9	416010.5938
2	10	290897.7187

Amounts

i	j	X ij
3	1	362031.9375
3	2	11635.8799
3	3	12408.5732
3	4	34679.8867
3	5	64468.1484
3	6	372031.0000
3	7	165875.3594
3	8	64440.3750
3	9	77543.0313
3	10	134885.8594

Amounts

i	j	X ij
4	1	33253.4414
4	2	6167.3862
4	3	129950.8828
4	4	265744.9062
4	5	1115.0125
4	6	46549.3477
4	7	2747.2842
4	8	259214.8907
4	9	360694.9687
4	10	94561.8906

Amounts

i	j	X ij
5	1	21680.6055
5	2	108742.3750
5	3	12363.8291
5	4	60274.5586
5	5	94179.5860
5	6	199626.3438
5	7	240523.6563
5	8	153777.9375
5	9	79971.0625
5	10	128860.0938

Amounts

i	j	X ij
6	1	1880.1201
6	2	20433.5918
6	3	25912.9141
6	4	26837.9375
6	5	494272.4063
6	6	11528.0469
6	7	119338.6797
6	8	37540.1836
6	9	18663.3535
6	10	7592.7417

Amounts

i	j	X ij
7	1	8863.7783
7	2	163225.7188
7	3	23243.9063
7	4	28646.9121
7	5	19476.0195
7	6	226819.0782
7	7	216548.2032
7	8	48374.8047
7	9	89028.3672
7	10	75773.2110

Amounts

i	j	X ij
8	1	41246.5352
8	2	45056.3125
8	3	37488.3125
8	4	47028.8633
8	5	20436.6641
8	6	95089.7110
8	7	60573.5508
8	8	107534.3360
8	9	7642.6030
8	10	1003.0950

Amounts

i	j	X ij
9	1	47224.0195
9	2	55099.3594
9	3	103608.0938
9	4	51547.1524
9	5	199471.0469
9	6	29462.9043
9	7	22964.9473
9	8	189129.4063
9	9	439.3862
9	10	1053.6534

Amounts

i	j	X ij
10	1	32359.2207
10	2	124174.0313
10	3	99595.2422
10	4	109363.9844
10	5	5074.2100
10	6	17714.3828
10	7	207875.8750
10	8	1579.4467
10	9	1715.7275
10	10	547.8546

Therefore 46232.0117 units should be shipped from warehouse 1 to customer 1 and so on down to 547.8546 units shipped from warehouse 10.

CONCLUSION

Multi stage simuation MSMCO has been tested on a variety of test problems (Conley 1994) in the literature. It is a versatile and useful technique on difficult multivariate problems (like the one presented here) when an approximate solution is acceptable. Additional examples of this general purpose simulation optimization solution technique are given by (Szarkowicz 1995) on an engineering problem, (Wong 1996) on applied mathematics problems and (Conley 2006) on packaging and real estate studies.
Multi stage Monte Carlo optimization can be considered as a possible solution choice for large scale nonlinear optimization problems, when other techniques are difficult or unavailable.

REFERENCES

Conley W.C. 1994. "Multi Stage Monte Carlo Optimization and Nonlinear Test Problems" International Joural of Systems Science, vol. 25, 155-171.

Conley W.C. 2006. "Optimization in Packaging and Real Estate" in Proceeding of the 2006 European Simulation and Modeling Conference (Toulouse, France, Oct 23-26), EUROSIS-ETI, Ostend, Belgium, 404-407.

Szarkowicz D.S. 1995. "Investigating the Brachistochrone with a Multi Stage Monte Carlo Method", International Journal of Systems Science, Vol, 26, 233-243.

Wong J.Y. 1996. "A Note on Optimization in Integers" International Journal in Mathematical Education in Science and Technology, Vol 27, 865-874.

BIOGRAPHY

William Conley is a professor of business administration and statistics at the University of Wisconsin in Green Bay, U.S.A. He has a B.A., M.A., M.Sc., and Ph.D. in mathematics and computer statistics. The developer of multi stage Monte Carlo optimization he has more than 195 publications worldwide. He is a senior member of Society for Computer Simulations International and was named to Who's Who in America and the Albion College athletic Hall of Fame in 1995 and 2005. He is a member of the American Chemical Society and a fellow in the Institute of Electronics and Telecommunication Engineers. He is a Michigan Scholar in College Teaching and was selected for the 2007 edition of Marquis Who's Who in the World. He teaches advanced business statistics to undergraduate students and does computer simulation research on multi stage Monte Carlo optimization and its applications.

INVESTIGATION OF THE SPATIAL DISTRIBUTION ALGORITHM OF THE TRAFFIC FLOW ANALYSIS AND OF THE ENTITY FLOW-PHASE ANALYSIS

Gábor Lencse
Department of Telecommunications
Széchenyi István University
Egyetem tér 1.
H-9026 Győr, Hungary
e-mail: lencse@sze.hu

László Muka
Elassys Consulting Ltd.
Bég utca 3-5.
H-1022 Budapest, Hungary
e-mail: muka.laszlo@elassys.hu

KEYWORDS

discrete-event simulation, traffic-flow analysis, entity flow-phase analysis, information and communication systems, business process systems

ABSTRACT

This paper investigates an important algorithm that is used in both the Traffic-Flow Analysis and the Entity Flow-phase Analysis. These methods are similar to each other and can be used for the fast and approximate (performance) analysis of Information and Communication Technology (ICT) systems and Business Process (BP) systems. Both methods contain an algorithm for the spatial distribution of the traffic (or entities) in the system. It is shown how the error of the spatial distribution can be measured, and the effect of the so called *size of routing unit* parameter of two algorithms is investigated.

INTRODUCTION

Performance Analysis Methods

Discrete-Event Simulation (DES) is a widely used method for the performance analysis (Jain 1991) of Information and Communication Technology (ICT) systems and Business Process (BP) systems. There are a large number of various methods used to describe the behaviour of complex systems (Banks et al. 1996; Bratley et al. 1986; Jávor 1985; Jávor 1993). The simulation of large and complex systems requires a large amount of memory and computing power that is often available only on a supercomputer. Efforts are made to use multiprocessor systems or clusters of workstations. The conventional synchronisation methods for parallel simulation (e.g., conservative, optimistic) (Fujimoto 1990) use event-by-event synchronisation and they are unfortunately not applicable to all cases, or do not provide the desirable speedup. The Statistical Synchronisation Method proposed by Pongor (Pongor 1992) does not exchange individual messages between the segments but rather the statistical characteristics of the message flow. This method can produce excellent speed-up (Lencse 1998) but has a limited area of application (Lencse 1999).

The fast (preliminary and approximate) performance estimation can be very useful in the early design state of an ICT or a BP system. We have proposed the Traffic-Flow Analysis (Lencse 2001) for the rapid performance estimation of ICT systems and the Entity Flow-Phase Analysis (Lencse and Muka 2006) for the fast investigation of BP systems.

Traffic-Flow Analysis

TFA is a combination of simulation and analytical and/or numerical methods. While the traditional discrete-event simulation models the travelling of each packet through the network, TFA uses statistics to model the networking load of applications. TFA works in two steps:

- In the first step, the method distributes traffic (the statistics) in the network, using the normal routing rules of the network.
- In the second part, the influences of the finite line and switching-node capacities are calculated.

The important features of TFA:

- The results are approximate but the absence or the place of bottlenecks is shown by the method.
- The execution time of TFA is expected to be significantly less than the execution time of the detailed simulation of the system.
- TFA describes the steady state behaviour of the network.

Entity Flow-phase Analysis

EFA has been derived from TFA by applying the TFA principles for BP systems. Methods of EFA (one-phase-method and multi-phase-method) are based on the same principles as TFA, only the interpretation of the model elements is different. The statistics represent entities (not messages) and the interpretation of the routing is also different. While the packets of a network usually do not multiply, the entities may fork (and the descendants must meet somewhere) or split (and the descendants live their own life separately); see more details in (Lencse and Muka 2006).

From now on we will focus on TFA, knowing that our results can also be applied for EFA.

Though it is not absolutely necessary, we encourage the reader of this paper to read the original paper on TFA (Lencse 2001) for the deeper understanding of the remaining part of this paper.

THE PROBLEM OF SPATIAL DISTRIBUTION

TFA is a general method, and can be used with any traffic model that satisfies the requirements of TFA for the traffic model. In the original paper, we proposed bit-throughput distribution and packet-throughput distribution (practically histograms) as traffic models to model the traffic on the lines and in the nodes, respectively.

The traffic model is always an *aggregated traffic model*, that is, it represents the complete traffic of a given type of applications that are connected to the given node. For example it represents the full traffic (in both directions) of 35 FTP applications that are connected to a router (by switches). If static routing is used, we can handle the complete traffic of the before mentioned 35 FTP applications (or 100 web browsers or any other type of applications) together: we must route only one statistics package through the network (containing the two types of histograms). However, if we have adaptive routing, then the traffic of a given type of application should not be handled together, rather it must be routed in multiple packets, each of which represent a given portion of the traffic of the given type of application connected to the given node. When determining the **size of the routing unit** (S_{RU}) we must consider the following issues:

The larger S_{RU} we choose, the fewer statistics packages are to be routed in the first phase and the less traffic model addition is to be performed in the second phase of TFA. However, if S_{RU} is too large, the spatial distribution of the traffic may considerably differ from the one that is formed in the detailed simulation of the system (and from the one in the real system). If S_{RU} is small, the spatial distribution of the traffic may be quite precise, but the larger amount of messages to be routed and traffic models to be added slow down the analysis. *The choice of S_{RU} must be a reasonable compromise (between the contradicting requirements) that is made in the knowledge of the whole system modelled.*

To be able to determine a good enough value for S_{RU}, we need to introduce a measure that expresses how good or bad a given spatial distribution of the traffic in TFA is, that is how well the given spatial distribution in TFA approximates the spatial distribution of the traffic in the detailed (packet-by-packet) discrete-event simulation of the system.

Before the presentation of the method that we propose for the good choice of S_{RU}, we introduce some formalism in the next sections.

FORMALIZATION AND INTRODUCING METRICS

The **capacity matrix** $\underline{\underline{K}}=[k_{ij}]$ describes the capacity of nodes and lines. The capacity matrix is an $n*n$ matrix, where **n** is the number of nodes in the network. Matrix element k_{ii} is the **routing capacity of node i** (measured in: packets per second), and matrix element k_{ij}, where $i \neq j$ is the **transmission capacity of the line** from node **i** to node **j**, measured in Mbit/s.
If there is no transmission line from node **i** to node **j** then $k_{ij} = 0$.

$$\underline{\underline{K}} = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,j} & \cdots & k_{1,n} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,j} & \cdots & k_{2,n} \\ \vdots & \vdots & & \vdots & & \vdots \\ k_{i,1} & k_{i,2} & \cdots & k_{i,j} & \cdots & k_{i,n} \\ \vdots & \vdots & & \vdots & & \vdots \\ k_{n,1} & k_{n,2} & \cdots & k_{n,j} & \cdots & k_{n,n} \end{bmatrix}$$

The **cost matrix** $\underline{\underline{C}}=[c_{ij}]$ defines the cost of communication through the network. Matrix element c_{ii} is the cost of the routing of a packet in node **i** and c_{ij} ($i \neq j$) is the cost of transmission of 1Mbit of information through the line from node **i** to node **j**. If there is no line from node **i** to node **j** then $c_{ij} = 0$.

The communication through the network is described by the **traffic matrix** $\underline{\underline{T}}=[t_{ij}]$. Matrix element t_{ii} describes result of TFA for node **i**: both the packet-throughput distribution and the delay distribution of TFA resulted in node **i**, and matrix element t_{ij} describes the result of TFA for the line from node **i** to node **j**: both the bit-throughput distribution and the delay distribution of the line from node **i** to node **j**.

For the evaluation of the results of the TFA distribution procedure, let us determine the **empirical load (utilization) matrix** $\underline{\underline{R}}=[r_{ij}]$, that gives us the simulation based load of every node and line:

r_{ii} = (the average number of packets for the node i)/ k_{ii}

r_{ij} = (the average number of Mbits for the line i→j)/ k_{ij} .

When different applications give us different matrices, then we can interpret the distance of these.

Let this be:

$$\left\| \underline{\underline{R}}_1 - \underline{\underline{R}}_2 \right\|_1 \text{ where } \|A\|_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}| \ .$$

This way for example, we can calculate whether the increased accuracy gained from the more detailed simulation is proportional to the increased processor time usage.

For the second phase of TFA we perform summation of traffic/load (see Lencse 2001). We may also decide to compare the resulting distributions.

In case of the comparison of distributions, it is advised to use the statistical distribution fitting method (χ^2 test, Hunyadi at al. 1996)).

Then the distance of the distributions is:

$$d = \chi^2 = \sum_{i=1}^{10} \left(f_i - g_i\right)^2 / f_i \text{ (grade of freedom 9)}$$

where g_i is the observed and f_i is the expected frequency for bin i.

Note that in case of d < 16,9 the distributions are considered equal, with confidence level of 95%.

For the evaluation of the results of the TFA, we introduce the **sample evaluation matrix** $\underline{\underline{S}}=|s_{ij}|$. Matrix elements may have the following values:

s_{ii}, s_{ij} = **3**, if the number of RUs through a given node or line < 30

s_{ii}, s_{ij} = **2**, if the number of RUs through a given node or line ≥ 30 but ≤ 200

s_{ii}, s_{ij} = **1**, if the number of RUs through a given node or line > 200

s_{ij} = **0**, if no line exists from node i to node j .

To compare results, we have to summarize all the elements of the $\underline{\underline{S}}$ matrix. The lower result is better because the number shows the level of uncertainty.

The **weighted sample evaluation matrix** $\underline{\underline{W}}=|w_{ij}|$ can be derived from matrix $\underline{\underline{S}}$ just multiplying the elements of $\underline{\underline{S}}$ by k_{ij}/c_{ij}. The meaning of this multiplication is that, in general, it is more important to have more precise results on large line or node capacities, and the increasing cost is decreasing the weight of a line or node.

The **alternative weighted sample evaluation matrix** $\underline{\underline{V}}=|v_{ij}|$ can be derived from matrix $\underline{\underline{S}}$ just multiplying the elements of $\underline{\underline{S}}$ by $1/(1-r_{ij})$. The meaning of this multiplication is that the less spare capacity we have, it is the more important to have more precise results.

A support matrix $\underline{\underline{B}}=|b_{ij}|$ may also be useful in the analysis. Matrix $\underline{\underline{B}}$ is a bitmap of the analysed network:

b_{ii}, b_{ij} = **1**, if there was RU travelling through the given node or line,

b_{ii}, b_{ij} = **0**, if there was not any RU travelling through the given node or line .

REVEALING THE ROUTING PROPERTIES

We examine a **data communication network** with **nodes** and **lines** between the nodes.

The aim of the routing is to transmit the information through the network with the **least cost and within the shortest time.**

TFA can be used with any routing method; the routing algorithm is the part of the network not of TFA.

Introducing Statistical Constraints

In statistics (Hunyadi at al. 1996) a sample consisting **N** elements is called as a **small** sample, when **N** < 30. Above a couple of hundreds the sample is a **large** sample and between these boundaries the sample may be looked as an **average** sample.
The results of simulation are reliable if the number of RUs travelling through a node or a line is at least a **several hundred** from an application.

According to considerations mentioned before, the S_{RU} should be determined as to generate *at least 200* statistical packages for a given type of application, otherwise the weight of coincidence would be too high, and the simulation results would not reflect the data or entity traffic on the network correctly.

Analysing the Routing Behaviour: Building the Routing Decision Tree

To model the decision process of the routing algorithm we use a decision analysis tool, the decision tree (Littlechild and Shutler 1991), and call it Routing Decision Tree (RDT).

We have to reveal the behaviour of the routing algorithm in the network.

Starting approach:

- If we have simulation results we may use it to construct the RDT
- If we have measurement results we may analyse it and then use the results to construct RDT .

Testing the Network

We make detailed simulation of the network during an appropriate test interval ($\mathbf{I_T}$) of the examination interval **T**. ($\mathbf{I_T}$ may be equal, for example, to 20% of **T** and may contain intervals considered to be typical.)
Important: the ***track*** (the sequence of nodes and lines for every application and every unit (packet/entity) sent by an application) for each application and for all of the units sent should be ***remembered.***

Now, we consider stopping criteria for the test:

Stopping criterion (S.1) for an application:

- If the number of units sent by an application through all of the lines on the routes of an application ≥ 200, or
- I_T has been spent .

Stopping criterion for the test process:

- Stopping criterion for all applications has been reached or
- I_T has been spent .

A *weak* stopping criterion may result in a less precise routing description.

Weak stopping criterion (S.2.1): stopping criterion based only on line data:

- Number of units from all applications taken together on all of the lines >200 .

Weak stopping criterion (S.2.2): stopping criterion based only on line data:

- Through some satisfactory portion (let us say 80%) of the lines the number of units sent > 200 and there are no *new lines* involved during the last significant inter-

val (let us say during the last 5% of I_T matrix $\underline{\underline{B}}$ is unchanged) of the test .

Weak stopping criterion (S.3): stopping criterion based only on applications:

- All of the applications have sent units > 200.

Differential stopping criterion (S.4) (may be combined with other criteria):

We build up the $\underline{\underline{R}}$ matrix:

- Send units
- Build $\underline{\underline{R}}_1$
- Send more units
- Build $\underline{\underline{R}}_2$
- Calculate the distance of $\underline{\underline{R}}_1$ and $\underline{\underline{R}}_2$
- If the distance < lower limit, then stop.

After the test phase, based on track data, we construct RDTs.

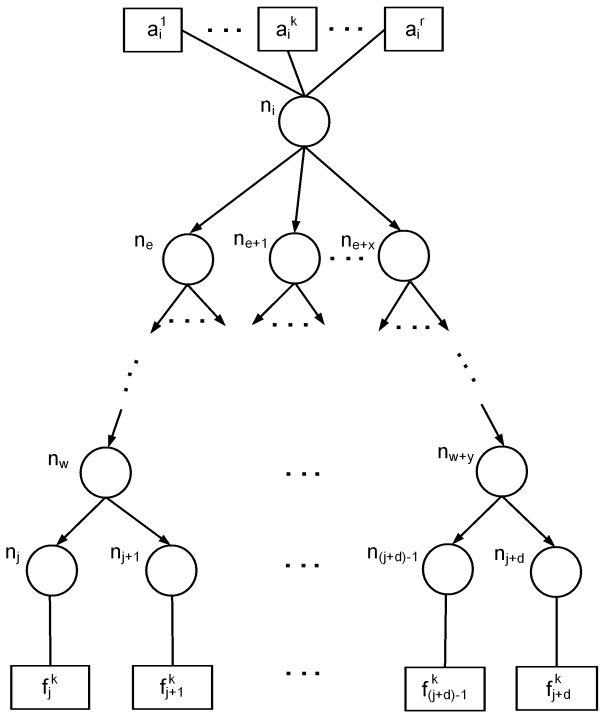


Figure 1. Routing Decision Tree

RDT in Figure 1 shows the probability (percentage) based decisions made by nodes on the route starting from the source node (n_i) to the destination nodes (n_j, …, n_{j+d}).

If there are parallel edges in the tree we split them and remember frequencies separately towards the destination (Figure 2).

In case of EFA, we may think about even the **replacing** of the original routing with RDT.

Based on the frequencies got from the test phase the S_{RU} for an application a_k may be calculated:

$$S_{RU} = N_T / \left(\sum_j f_j^k \right) \left(\min f_j^k / 200 \right), \text{ where}$$

a_i^k is an application connected to node **i**.

N_T is the quantity of information (packets for nodes, Mbits for lines) sent by a_i^k during T.

f_j^k denotes the measured destination frequency (from a_i^k to node **j**).

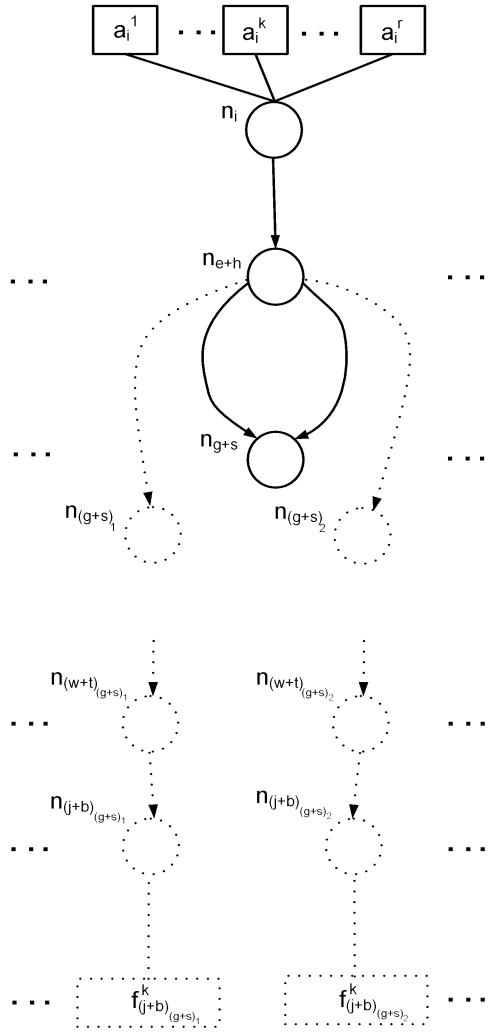


Figure 2. Splitting Edges in Routing Decision Tree

PERFORMING SPATIAL PHASE

Using S_{RU}s, calculated in the previous way, we perform the spatial distribution phase of TFA.

About Dynamic Control of S_{RU}: Increasing-Decreasing Decisions During Spatial Phase

Increase S_{RU}: during the last significant part of distribution there was no change in the set of lines (routing seems to be static) for all of the applications or for one application.

Decrease S_{RU} one: a new line occurred in the set of lines (there was a change in any element of matrix $\underline{\underline{B}}$). The new line has to be inserted into the RDTs and a new S_{RU} should be calculated.

Decrease S_{RU} two: in the end if *the number of RUs < 200* for a line or a node we may decide to recalculate S_{RU}s and repeat the process.

We may also consider using RUs with different sizes on different lines. For example, it may be useful to use smaller RUs in the case of lines in critical or overloaded state. To use smaller RUs instead of the ones that arrived to a given line, a **RU conversion** should be made before entering the line.

If we need to use **smaller** RUs than we have, we generate smaller RUs (for example with exponential distribution) that together represent the same amount of traffic as the original RU represented.

If we need to use **larger** RUs then we have, some smaller RUs are replaced by a larger one. Of course, we can do it only if the smaller ones are present together at a given point of the network, for example waiting in a queue.

Evaluation of Results

If we have detailed simulation results then using $\underline{\underline{R}}$ we may compare TFA results to detailed simulation results: the closer the results are to detailed simulation results the better the TFA results may be considered.

The sample evaluation matrix $\underline{\underline{S}}$ (together with $\underline{\underline{W}}$ and $\underline{\underline{V}}$) may be used to analyse the reliability of results.

CONCLUSIONS

We have introduced formal description for the networks and traffic conditions such as: capacity, cost, traffic and utilization matrices as well as metrics for the difference of the traffic and utilization matrices.

On the basis of the statistical constraints on sample size, we have introduced the sample evaluation matrix, the elements of which express if the number of RUs are high enough for a given node or line.

We have given a method, how to calculate the S_{RU} by using the Routing Decision Tree and detailed simulation for an appropriate (short) period of time. We have also given different stopping criteria for the simulation.

We have shown how the S_{RU} can be dynamically controlled during the spatial distribution phase of TFA or EFA.

We conclude that with our results on the appropriate choice of the S_{RU} of TFA or EFA, these methods have been matured for implementation.

REFERENCES

Banks, J.; J. S. Carson; B. L. Nelson. 1996. *Discrete-Event System Simulation* Prentice Hall, Upper Saddle River, New Jersey

Bratley P.; B. L. Fox; and L. E. Schrage. 1986. *A Guide to Simulation.* Springer-Verlag, New York

Elassys Consulting Ltd. 2007. *ImiNet and ImiFlow Systems* http://www.elassys.hu

Fujimoto, R. M. 1990. "Parallel Discrete Event Simulation" *Communications of the ACM* 33, no 10, 31-53

Hunyadi, L.; Mundruczó, Gy.; Vita, L. 1996. *Statisztika* Aula Kiadó, Budapest

Jain, R. 1991. *The Art of Computer Systems Performance Analysis.*. John Wiley & Sons, New York

Jávor, A. (editor) 1985. *Simulation in Research and Development.* North-Holland, Amsterdam

Jávor, A. 1993. *Petri Nets in Simulation* EUROSIM Simulation News Europe, 1993, no. 9, pp. 6-7.

Lencse, G. 1998. "Efficient Parallel Simulation with the Statistical Synchronization Method" *Proceeding of the Communication Networks and Distributed Systems Conference* (CNDS'98), (San Diego, CA, USA, January 11-14) SCS, 3-8

Lencse, G. 1999. "Applicability Criteria of the Statistical Synchronization Method" *Proceedings of Communication Networks and Distributed Systems Conference* (CNDS'99), (San Francisco, CA, USA, January 17-20.) SCS, 159-164

Lencse, G. 2001. "Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems" *Journal of Computing and Information Technology* 9, No. 1, 15-27.

Lencse, G., Muka, L. 2006. "Expanded Scope of Traffic-Flow Analysis: Entity Flow-Phase Analysis for Rapid Performance Evaluation of Enterprise Process Systems" *Proceedings of the 2006 European Simulation and Modelling Conference* (ESM'2006) (Toulouse, France, 2006. Oct. 23-25.) EUROSIS-ETI, 94-98.

Littlechild, S. C.; Shutler, M. F. 1991. *Operations Research in Management* Prentice Hall, London

Pongor, Gy. 1992. "Statistical Synchronisation: a Different Approach to Parallel Discrete Event Simulation" *Proceedings of the 1992 European Simulation Symposium* (ESS'92), (Dresden, Germany, Nov. 5-8) SCS Europe, 125-129.

BIOGRAPHIES

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994 and his Ph.D. in 2000. The area of his research is (parallel) discrete-event simulation methodology. He is interested in the acceleration of the simulation of info-communication systems. Since 1997, he works for the Széchenyi István University in Győr. He teaches computer networks and networking protocols. Now, he is an Associate Professor. He is a founding member of the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University. He does R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998. Dr Lencse works part time at the Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005. There he teaches computer architectures.

LÁSZLÓ MUKA graduated in electrical engineering at the Technical University of Lvov in 1976. He got his special engineering degree in digital electronics at the Technical University of Budapest in 1981, and became a university level doctor in architectures of CAD systems in 1987. Dr Muka finished an MBA at Brunel University of London in 1996. Since 1996 he has been working in the area of simulation modelling of telecommunication systems, including human subsystems. He is a regular invited lecturer in the topics of application of computer simulation for performance analysis of telecommunication systems at the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University of Győr.

An Integrated Environment for Urban Traffic Flow Simulation Using Cellular Automata

Ourania Hatzi
Stephanos Thomas
Vassilis Dalakas
Mara Nikolaidou
Dimosthenis Anagnostopoulos

Harokopio University of Athens
70 El. Venizelou Str, 17671
Athens, GREECE.

Keywords: Urban Traffic Flow Modelling, Cellular Automata, Visual Environment, MATLAB, Java

Abstract

Simulation of large traffic networks is becoming an essential tool for traffic studies as the number of vehicles increases continuously and traffic conditions deteriorate, both for vehicles and pedestrians. This paper discusses the development of a visual environment for micro-simulation of vehicle traffic flow in road networks and highways, employing cellular automata. The integrated tool targets at non-expert users who wish to study urban traffic and not smulationists, thus we focus on parameterization capabilities and visualization features. The environment copes with expandability and parameterization of both the road networks and the simulation rules. Our goal is to facilitate the construction of hierarhical modular traffic models, provide the ability to experiment with alternative simulation rulesets and offer efficient viualization, which accomodates monitoring traffic conditions and exporting useful conclusions.

1. INTRODUCTION

Currently, the simulation of traffic flow in urban networks is becoming an important factor not only in transportation studies, but in environmental and safety studies as well [4-6, 13]. A variety of models for traffic flow has been developed by scientists from several areas, each coping with the problem from a different perspective. In many cases, corresponding simulation tools employing these models have been implemented. However, as changes in vehicle technology occur, new road network structures appear and pedestrian and driver behaviour alters over time, the simulation models for traffic flow of the past may have to be adjusted to reflect real world conditions more accurately. Such adjustments would also serve the exploration of traffic flow on a localized basis, where classical single-lane models might not be accurate enough. Furthermore, most traffic management projects require simulation study before realization, in order to be cost-effective. Therefore, great interest lies in the development of expandable simulation tools that should be capable of encapsulating the complexity of contemporary urban complex environments, while at the same time should be fully paremeterizable in order to easily adapt to real world changes.

A research in literature shows that so far, there is a variety of tools, such as TRANSIMS [7], which facilitate the parametric definition of road network models. However, model expandability and parametric composition is not provided, while simulation rules are embedded within the model and cannot be modified.

In this paper an integrated environment incorporating cellular automata is proposed, accomodating the definition and study of hierarhical, modular, parameterizable models, and facilitates the modeller's intervention not only to road network models but to simulation rules as well. In addition, the environment aims at a high degree of integration by including visualization features. These features include efficient visualization of simulation results, and comparison with empirical results in charts. Moreover, an important aspect of the environment is the real-time visualization of certain traffic measures, which facilitates monitoring of traffic conditions in urban networks and detecting specific conditions such as congestion. Such features are also exploited in safety studies, where efforts to locate the most appropriate places for pedestrian crossings take place. The environment is developed on two different platforms which cooperate, in particular MATLAB for the simulation modules and Java for the visualization modules.

The rest of the paper is organised as follows: Section 2 reviews the Cellular Automaton Model, Section 3 presents the features of our environment, as well as validation issues, and Section 4 discusses future work and goals.

2. THE CELLULAR AUTOMATA MODEL

The proposed approach was based on the cellular automaton model for freeway traffic proposed by Kai Nagel and Michael Schreckenberg [1,2]. The model is

defined by a vector which is comprised of sites, also known as *cells*. The size of each cell is usually set to 7.5 m. Each cell may either be empty, or occupied by a vehicle. In the original model, all vehicles are of the same size and have integer velocities between zero and u_{max}, which is a parameter of the model. An ordinary, realistic value for u_{max} is 5. Update of the vehicle position involves three rules, which, at each time step, determine the velocity of the vehicle. These rules are the following:

- *Acceleration*: if the velocity of the vehicle has not reached the maximum velocity u_{max}, it is increased by 1.
- *Slowing down*: if the gap between a vehicle and its preceding vehicle, counted in cells, is less than the velocity of the vehicle, the velocity is descreased to gap-1.
- *Randomization*: With a probability pdec, the velocity of a vehicle is decreased by 1.

The above rules are executed and velocity is calculated for each vehicle present in the system. Eventually, the position of each vehicle is updated according to this velocity. In order for the results to correspond to real-life measurements, each time step is considered to be 1 sec.

In order to demonstrate the effect of the above rules to the velocity of the vehicles, the following example is presented, involving a road comprised of twelve cells. The first, the fourth and the last cell are occupied by vehicles with velocities 2, 5 and 3, respectively. The pdec for this example is set to 0.33, therefore, an average of 1 out of 3 vehicles will be forced to reduce its velocity. Applying the rules will result in the first vehicle increasing its velocity by 1 but then decreasing it again because of the small gap of only two cells between the first and the second vehicle. The second vehicle has already reached u_{max} therefore it will not further increase its velocity. In case randomization requires this vehicle to decelerare, its velocity will be reduced to 4. Finally, the third vehicle has a velocity that allows it to leave the system. The updates in the vehicle positions will take place, simulation time will advance and the next simulation step will begin by applying again the same rules.

The most important measures in the model are traffic flow and traffic density, which, on a single lane, are computed from the local measurement data. The traffic flow is defined as the number of vehicles per time unit and is calculated as follows:

$$q_j(x,t) = \frac{\Delta N_j(x,t)}{\Delta t}$$

The mean traffic density is defined as the number of vehicles per length unit and can be calculated approximately from the traffic flow and the mean velocity:

$$\rho_j(x,t) = \frac{\Delta N_j(x,t)}{\Delta x_j} \approx \frac{\Delta N_j(x,t)}{\Delta t \cdot v_j(x,t)} = \frac{q_j(x,t)}{v_j(x,t)}$$

The approach $\Delta xj \approx \Delta t$ □ vj (x, t) is valid only if all detected vehicles are moving with approximately the same velocity during the time interval Δt.

3. INTEGRATED SIMULATION ENVIRONMENT

The proposed tool aims at providing enhanced visualization and parameterization capabilities for modeling and simulating urban traffic. The cellular automata model, described in the previous section, is frequently used for modeling traffic flow. Evidently, experts in this field would appreciate an efficient and easy to use visual interface for conducting simulation experiments. The proposed simulation environment aims at providing such a visual interface, and accomodating the expandability and parameterization of road network models and alternative simulation rules. Scalability of the road networks is not an issue, as road networks can be arbitrarily large and new parts can be progressively added, at any time, to the existing road networks.

Simulation result exploitation and comparison features are supported through a distinct module, while monitoring of traffic conditions in each separate part of the road network in real time is also provided. Among the other elements of road networks available for monitoring, there are traffic lights and pedestrian crossings, which are of great significance when it comes to pedestrian safety studies.

The modular architecture of the environment is illustrated in Fig. 1. Most of the modules were implemented in MATLAB, while for the visualization modules, Java was considered more appropriate.

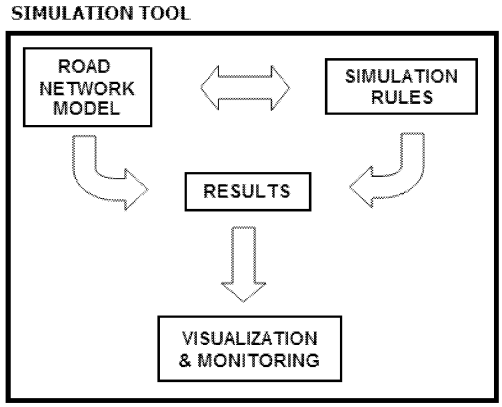


Figure 1. Integrated environment architecture.

The simulation environment facilitates:

- Parametric definition of road network models.
- Easy modification and expansion of existing road network models.
- Adaptive ruleset that can easily be altered.
- Parametric simulation.
- Visualization of simulation results in charts and comparison with empirical data.
- Real-time visualization of simulation measures and traffic conditions in road network.

3.1. Parametric road network definition

The conceptual model of the road network, as designed in the proposed system, consists of roads and junctions. Each road is divided into lanes and parts, and each part into cells. Junctions may be simple, in which case regular traffic laws indicate priority of vehicles, or may have traffic lights which concern vehicles, pedestrians or both, in which case their indications are taken into account. Moreover, the system has entry and exit points, where vehicles enter and leave the system and statistics are gathered. As far as properties of the road parts are concerned, each part has a predefined capacity, which indicates the number of cells, and a deceleration factor, which indicates how often the vehicles slow down in this part of the road in a probabilistic way. The deceleration factor accounts for unpredictable events that may cause a vehicle to slow down, such as unexpected pedestrian behaviour. To demonstrate the model, a simple road network is presented as an example in Fig. 2.

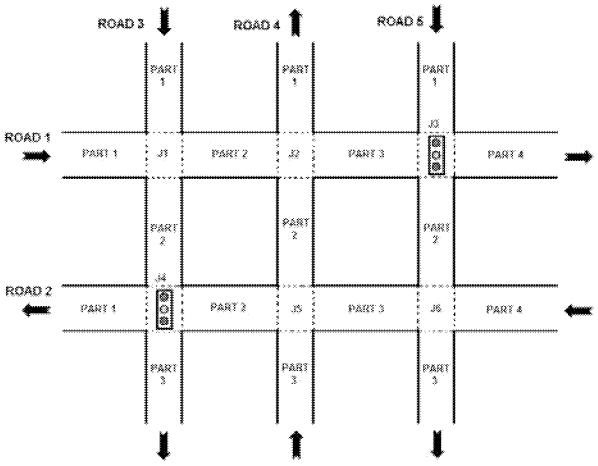


Figure 2. An example road network with entry and exit points marked and its representation in terms of cellular automata.

Each vehicle is represented as a separate entity which has its own parameters, such as velocity. These parameters are used in combination with the simulation rules and other safety parameters of the system, in order to produce the updates to the position of the vehicle. Vehicles are inserted in the system at entry points, according to a predefined arrival rate given by the user in advance as a parameter of each entry point. While they are in the system, they can occupy exactly one cell at any given time, as they are all considered to be of the same size. Finally, they leave the system at exit points, affecting the appropriate statistic measures.

The implementation of the aforementioned model in MATLAB involves a combination of arrays and other variables. The structural information of the road network is provided either through user interaction or formatted text files.

The network consists of five single-lane roads and junctions between them, two of which have traffic lights. The direction of the roads as well as the entry and exit points of the network are also marked. The network is divided into roads, parts and junctions as indicated.

A partial example demostrating the insertion of information for the first part of Road 1 to MATLAB follows (Fig. 3):

```
How many roads are there in the network?  5
How many parts do you have in road 1?  4
Is part/lane 1 an input point for the network?
(no=0 / yes=1):  1
Enter interarrival time (lambda):  1.5
Enter part/lane 1 capacity?  50
Which part/lane is on the right of part/lane 1
(0 for no part/lane):  0
Which part/lane is on the left of part/lane 1
(0 for no part/lane):  0
What is the type of the intersection at the end
of this part/lane?
(roundabout=0, traffic light=1, dead end=2,
other lane=3, network end=4):  3
What is the id of the intersection at the
beginning of this part/lane?  0
What is the id of the intersection at the end
of this part/lane?  1
```

Figure 3. Example of road data input through user interaction.

The information provided to the system includes the interarrival time for this part, since it is an entry point for the system, the capacity of the road part, its direction, its neighbouring lanes, if any, and the junctions at the beginning and end of the road part. Traffic light information is provided in a similar way. The same procedure has to be followed once for every part and every traffic light in the system. Then, it can be saved and modified at a later time, used to run a traffic simulation with the simulation rules embodied in the system, or used with a different set of simulation rules.

It should be noted that the road network model is saved as a simple text file in order to be easily expanded and modified. An example of such a file for the above data input follows (Fig. 4):

```
1     1  1 1.50  50  0  0 E   1   0   1
1     2  0 0.00  50  0  0 E   4   1   2
2     1  1 2.10  35  0  0 S   1   3   1
2     2  0 0.00  40  0  0 S   4   1   4
```

Figure 4. Example formatted file for road network data storage.

For example, if the modeller wishes to add a crosswalk in part 3 of road 2, this part can be splitted into two parts, and the new information can substitute the old data in the text file, without any further changes. Likewise, new roads or road parts can be added to the network by appending new data to the text file.

3.2. Simulation

Each step of the simulation consists of three sub-steps:

- Importing the vehicles
- Updating vehicle positions
- Advancing the simulation time

The above steps comprise different sub-modules of the simulation module. Therefore, each one is independent from the other, thus enabling the modeller to experiment with different sets of simulation rules.

The implemented model in the proposed system is the original one by Nagel and Schreckenberg. The same simulation rules and conventions for cell length and maximum speed are followed. At each time step, the rules that determine the velocity of each vehicle are acceleration, slowing down and randomization, and in the end the position of the vehicle is updated according to this velocity. Randomization probability, referred to as pdec, is introduced in the system as a parameter of each road part, therefore each part may have a different probability.

The above rules are encoded as a module in the system and executed consecutively. As long as a vehicle stays in the same part of the road, updating its position is a trivial task. However, attention must be paid to take the appropriate actions when a vehicle approaches the end of the road part it is moving in, and its velocity enables it to leave this road part. Priority in junctions and traffic light indications must be taken into account. Moreover, there is the possibility that the vehicle has reached a network exit point, therefore statistics must be updated.

3.3. Visualization

At the end of each simulation step, measures of interest for each road part and for the network in general are calculated and exported to a formatted text file for further utilization. Some of these measures, such as vehicle density, enable monitoring and exporting conclusions about traffic conditions in road parts. Therefore, some kind of real-time visualization is considered necessary in order to accomodate the study of critical properties of urban traffic, both for vehicles and pedestrians. This is achieved through another module which provides visualization of the road network along with real-time visualization of appropriate measures. The road network is presented as a map, preserving the spatial properties of road parts, such as direction and capacity, which makes the interface familiar to the users, while the measure values are depicted using indicative colours, with each colour representing a range of values. This level of abstraction, facilitated by the use of intervals instead of specific values, enables the identification of congestions and other abnormal traffic conditions at a glance, even for inexperienced users. Apart from real-time monitoring of traffic conditions and detecting temporary abnormalities, the visualization can also serve as long-term monitoring of traffic behaviour over a period of days or weeks. In this case, decision making can be facilitated as far as traffic lights and pedestrian crossings are concerned.

The visualization module was not implemented in MATLAB, but instead the Java platform was preferred. This choice is justified by the graphical and interface capabilities of Java, the expandability of the language, as well as the ability to export applications on the web in the form of applets or web services. An example of the interface in presented in Fig. 5. The road network represented includes the roads approximate to Harokopio University in Athens, and the measure of interest here is vehicle density per road part. In this example only three intervals were used, and the colours representing them are green, yellow and red. As the simulation evolves, new values of the measures are calculated in each simulation step, therefore the colours in each road part change.

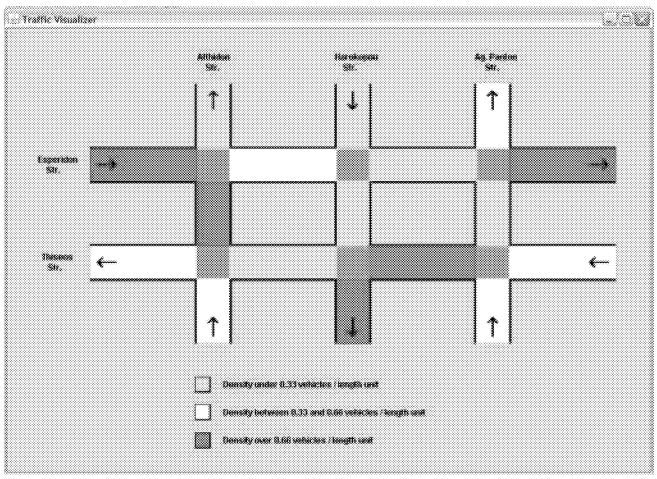


Figure 5. Real-time visualization of vehicle density in an indicative road network.

3.4. Validation and results

In order to facilitate validation, the tool offers the ability to compare simulation results to empirical results using charts. However, a series of steps has to precede comparison: First of all, road networks have to be modelled and simulated with the simulation ruleset of choice. The default ruleset is the one by Nagel and Schreckenberg, which accomodates simple urban traffic in the original cellular automata model. However, as the simulation rules comprise a different module, they can easily be altered without affecting the rest of the tool. In addition, numerical results have to be produced during simulation in order to be compared to empirical results, and empirical results themselves must be gathered and inserted in the system.

The example results in Fig. 6 concern the simulation of a simple network that consists of one road with two parts. Simulation was performed for different values of the probability factor pdec, which serves randomization purposes and shows how often vehicle velocity is decreased, and the chart presented concerns the value 0.2. The measure of interest in this example is the number of

vehicles that went through the network in each simulation step.

Taking into account the results from the simulation of simple models and their proximity to empirical results, the reliability of the proposed tool was established. Thus, further experimentation with more complex traffic models and different simulation rulesets can be explored.

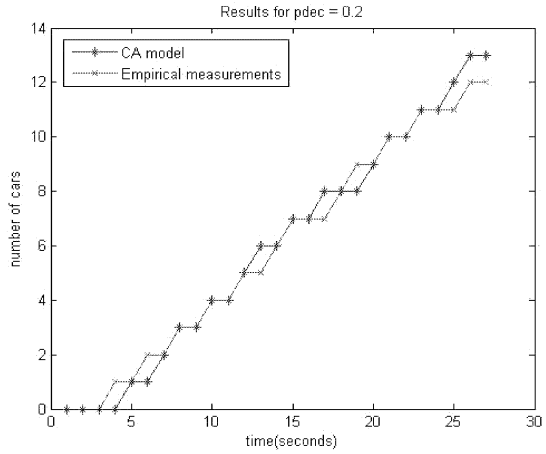


Figure 6. Comparison of simulation and empirical results.

4. CONCLUSION AND FUTURE WORK

The proposed tool accomodates the definition and experimentation of hierarhical modular models and simulation rules in an independent fashion, along with efficient visualization of results and real-time visualization of certain traffic measures. We are currently working on a case study of an autonomous inner city network in downtown Athens, in order to demonstrate the use of the tool in more complex urban environments and with altered, localized simulation rules, which ensure accuracy.

The next steps include the ability to incorporate other types of metrics in the system, such as environmental ones, and incorporate these metrics in the visualization module. Moreover, the possibility of combination of vehicle and pedestrian traffic in a single simulation tool is explored. This necessitates pairing simulation rules from both fields [12], therefore extensions to the modules of the tool are required. This combination is estimated to lead to more complete conclusions as far as safety studies are concerned.

Acknowledgments

This research was supported in part by Pythagoras program (MIS 89198) co-funded by the Greek Government (25%) and the European Union (75%).

References

[1] von Neumann, J. 1966, *The Theory of self reproducing automata.* Edited by A. Burks, Univ. of Illinois Press, Urbana

[2] Nagel K and Schreckenberg M. 1992, "A cellular automaton model for freeway traffic". J. Physique I 2 2221

[3] Knospe W., Santen L., Schadschneider A. and Schreckenberg M. 2001, "Single-vehicle data of highway traffic:microscopic description of traffic phases". Phys. Rev. E at press

[4] 58th Traffic and Safety Conference, May 9-11,2007, Columbia, Missouri. *Available online via http://muconf.missouri.edu/traffic/index.html [accessed April 19, 2007]* .

[5] Ontario Traffic Conference, Ontario. *Available online via http://www.otc.org/otcweb2 [accessed April 19, 2007]* .

[6] European Transport Safety Council. *Avalable online via http://www.etsc.be/home.php [accessed April 19,2007]* .

[7] TRANSIMS, *The TRansportation ANalysis and SIMulation System project.* Los Alamos National Laboratory.

[8] Simon P. M. and Nagel K. 1998, "Simplified cellular automaton model for city traffic". Phys. Rev. E 58, 1286 – 1295

[9] Fouladvand M. E., Shaebani M. R., Sadjadi Z., 2004, "Intelligent controlling simulation of traffic flow in a small city network". Journal of the Physical Society of Japan, vol. 73, no11, pp. 3209-3214.

[10] Ping-Hua H., Ling-Jiang K., Mu-Ren, L. 2002 "A study of a main-road cellular automata traffic flow model". Chinese Physics Beijing Journal, VOL 11; PART 7, pages 678-683.

[11] Barlovic R., Esser J., Froese K., Knospe W., Neubert L., Schreckenberg M., Wahle J., 2001, "A cellular automaton traffic flow model for online simulation of traffic". Parallel Computing, Volume 27 , Issue 5, Special issue on cellular automata: from modeling to applications, Pages: 719 – 735.

[12] Blue, V.J., Embrechts, M.J., Adler, J.L. 1997, "Cellular automata modeling of pedestrian movements". IEEE International Conference on Systems, Man, and Cybernetics 1997, 12-15 Oct 1997 Page(s):2320 – 2323 vol.3

[13] Environmental Impact of Transportation Conference. *Available online via http://www.planeta.com/ecotravel/tour/transportation.html [accessed August 3, 2007]*

ADAPTATION AND SELF-ORGANIZATION APPLICATIONS

TIME OPTIMAL ADAPTIVE TARGET TRACKING BY A MOBILE ROBOT

Krzysztof Skrzypczyk
Department of Automatic Control
Silesian University of Technology
Akademicka 16, 44-100, Gliwice, Poland
E-mail:krzysztof.skrzypczyk@polsl.pl

KEYWORDS
navigation, adaptation, prediction, time optimality, tracking

ABSTRACT

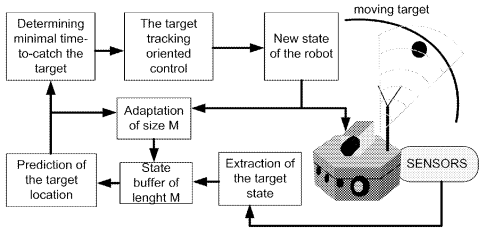
In this paper a method of navigating a robot toward moving target is discussed. Time optimal solution of target tracking problem is presented. The proposed method is very effective and can be applied even to the very simple processing unit. Fast adaptive algorithm of prediction as well as the time optimal algorithm of tracking are presented in the paper. Results of simulations are presented to prove an efficiency of the proposed approach.

INTRODUCTION

Tasks a mobile robot is considered to perform are mainly related to a problem of moving a robot from an initial location to a desired one. This problem has been a subject of very intensive research for last 40 years. The problem is getting more complicated when an assumption is made that the target of navigation is not static. This kind of problem is often called a tracking one (Liao et al. 2003; Gass 2005). Moreover multi robot domains very often requires collaboration between robots and then the problem of tracking a movement of another team-mate is relevant. Solving such a problem needs two main issues have to be taken into account: prediction of a movement of the target as well as fast tracking algorithm. The importance of trajectory prediction and its influence on a quality of planning of a motion of the robot is rather obvious. The better estimate of a future location of the target is obtained the more effective tracking trajectory can be planned. The problem of prediction of trajectory is well represented in the literature. The most popular approach to a solution of this problem is Kalman Filtration (Åström and Wittenmark 1997) and probabilistic approach called particle filter (Gustafsson et al. 2002). Unfortunately these approaches give only good, short term prediction of the trajectory. Moreover if an application of these algorithms to multi agent systems is considered they must be simple and non complex. In the paper (Skrzypczyk 2006) a simple algorithm of prediction was presented. Although it is very effective it lacks the feature of adaptation and in some cases can be a bit difficult to tune properly. In this paper a new adaptive prediction algorithm based on the features of aforementioned algorithm (Skrzypczyk 2006) is presented. Thanks to ability of adaptation is more effective and robust in comparison to the original one. Another problem is related to tracking a given trajectory. If the trajectory is known the task is rather simple and there exist many methods of solution of the problem. But if the tracking process is based only on an estimation of the target trajectory the problem is getting more complicated especially due to errors of the prediction method or (and) unpredictability of the target movement. Additionally when there is a need of minimizing the time of tracking the problem is getting more and more complicated. In this paper a heuristic approach to the time optimal tracking problem is presented. An algorithm that utilizes some common-sense knowledge and prediction technique is discussed. The simlational evaluation of this approach is presented in the further part of this paper.

The System Architecture Overview

Figure 1 presents a general diagram of the robot control system. The robot is assumed to have a sensory system that is able to provide information about a location of the tracking target. These data are stored in a *state buffer* of length *M* and are used to build a target trajectory model. The key issue in the proposed algorithm of prediction is a method of adaptation of the length *M*. The number of past data has a great influence on effectiveness of the prediction. Using the model the predicted location of the target is computed and minimal time-to-reach the target is determined. Next stage is to plan tracking trajectory of the robot. During an execution of the motion plan the errors of prediction are monitored. In case when the tracking error becomes greater then some threshold value the procedure is repeated with new adapted parameters.



Figures 1: A General Diagram of the Robot Control System

PROBLEM FORMULATION

To specify the problem let us introduce the notation that will be used hereafter. Let us denote the state of the robot by a vector $X(t_n) = [x_r(t_n), y_r(t_n), \Theta_r(t_n)]$ where its elements define coordinates of the location of the robot and its

orientation measured in a moment t_n of discrete time. The control of the robot in a moment t_n of discrete time is defined as a vector $u(t_n) = [\omega(t_n), v(t_n)]$. The target moves along a trajectory a model of which is unknown to the robot control system. The location of the target in a moment t_n is denoted by . $P(t_n) = [x_T(t_n), y_T(t_n)]$. Now the problem can be stated as follows: Find the control of the robot that allows to reach the target in the shortest possible time $t_{\min} = H_{\min}\Delta t$ $H_{\min} = 1, 2, \ldots$. So we look for the control vector $U^* = [u(t_n), u(t_{n+1}), \ldots, u(t_{n+H_{\min}-1})]$ that provides $\|X(t_{\min}) - P(t_{\min})\| \le \varepsilon_T$ where the distance is measured according to Euclid norm and the ε_T is an accuracy of tracking.

ADAPTIVE PREDICTION METHOD

In the work (Skrzypczyk 2006) a predictive algorithm of tracking the robot was developed. This method gives *H* step prediction based on *M* past observations of the state of the moving object. Unfortunately it lacks the ability of adaptation of the length *M* of the vector of observations. Here in this work an improvement of the method is presented. The algorithm of adaptation starts with setting initial values. The n_0 denotes the current discrete moment of time. The M_{max} is a maximal possible number of past data the prediction model can be built on. The $\varepsilon(n_0)$ is initial value of the one step prediction error, usually set to value of 0. Next step of the algorithm is to compute the prediction model. In next (*n=n+1*) discrete moment of time the error of prediction is calculated according to:

$$\varepsilon(n) = \left\| P(n) - \hat{P}(n_0 + 1) \right\| \tag{1}$$

If the error (1) is greater then some threshold value ε_0 that means the rapid change of movement has happened and the model is no longer valid. The *M* is set to the value of 1. In other case it is checked if the current error $\varepsilon(n)$ has changed considerably. In order to check it the following mean error index is used:

$$\delta(n-1) = \frac{1}{M}\sum_{m=1}^{M} \varepsilon(n-m) \tag{2}$$

The (2) is compared to the error $\varepsilon(n)$. If the difference is not significant that means there is no reason to increase *M*. In other case it must be checked whether to increase or decrease the number *M*. In order to do that the relation $\delta(n-1)\big|_M < \delta(n-1)\big|_{M-1}$ is checked. If value of the (2) calculated for the greater value of *M* is lower than calculated for the lower value of *M,* then the *M* is increased by 1. Otherwise it is decreased by 1. Although the method seems to be very simple the multiple simulations proved its efficiency. The algorithm provides satisfactory prediction and reduce the number of *M* to the range $M \in \langle 3,5 \rangle$.

TIME OPTIMAL TRACKING

To provide time optimal tracking the value of t_H must be minimal. The problem of time minimal control is well known

---

Step 1: Set $H_{\min} = 1, t = H_{\min}\Delta t$

Step 2: Calculate the speed of the robot after the time *t*, according to:

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} v_0 + a_v t \\ \omega_0 + a_\omega t \end{bmatrix} \tag{15}$$

Step 3: Check if $v(t) < v_{\max} \wedge \omega(t) < \omega_{\max}$.

Step 3.1: If yes, calculate the translation and the rotation of the robot according to:

$$\begin{bmatrix} s(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} v_0 t + 0.5 a_v t^2 \\ \omega_0 t + 0.5 a_\omega t^2 \end{bmatrix} \tag{16}$$

Go to the step 4th.

Step 3.2: If no, calculate the translation and the rotation of the robot according to:

$$\begin{bmatrix} s(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} v_0 t_{v\max} + 0.5 a_v t_{v\max}^2 + (t - t_{v\max}) v_{\max} \\ \omega_0 t_{\omega\max} + 0.5 a_\omega t_{\omega\max}^2 + (t - t_{\omega\max}) \omega_{\max} \end{bmatrix} \tag{17}$$

Go to the step 4th.

Step 4: Check if $\alpha(t) < \Delta\Theta(t) \wedge s(t) < L(t)$.

Step 4.1: If yes, $t_{\min} = t = H_{\min}\Delta t$.

Go to the step 5th.

Step 4.2: If no, $H_{\min} = H_{\min} + 1$, $t = H_{\min}\Delta t$.

Go to the step 2nd.

Step 5: Stop

Figures 2: The Iterative Algorithm of Determining the Minimal Time Horizon.

and there are many successive solutions of the problem in case the trajectory of the target is known. Moreover even in such a case the solution methods are quite complex what implies difficulties in their implementation. Therefore we found reasonable to revisit the problem and to find as simple as possible solution of the problem for a small nonholonomic robot.

Estimation of Minimal Time Horizon

Let us start with introducing some notation that will be used hereafter. The time t_0 is the one the process of tracking starts in. The v_0, ω_0 are values of the linear and the angular velocity the robot moves with in the moment t_0. The $\Theta_d(t_0 + H_{\min}\Delta t)$ is a desired heading of the robot necessary to reach the target in the given time. The value:

$$\Delta\Theta(H_{\min}\Delta t) = \left| \Theta_r(t_0) - \Theta_d(t_0 - H_{\min}\Delta t) \right| \tag{3}$$

is the increment of heading of the robot necessary to reach the target in the minimal time t_{min}. Similarly are defined notions concerning linear translation of the robot:

$$L(H_{\min}\Delta t) = \left\| X(t_0) - \hat{P}(t_0 + N_{\min}\Delta t) \right\| \tag{4}$$

The distance (4) is calculated according to Euclid norm. The time $t_{\omega\max}$ and $t_{v\max}$ is the time the robot needs to reach correspondingly its angular and linear maximal velocity. They are calculated according to:

$$t_{\omega\max} = (\omega_{\max} - \omega_0)/a_\omega, t_{v\max} = (v_{\max} - v_0)/a_v \tag{5}$$

Target Driven Control

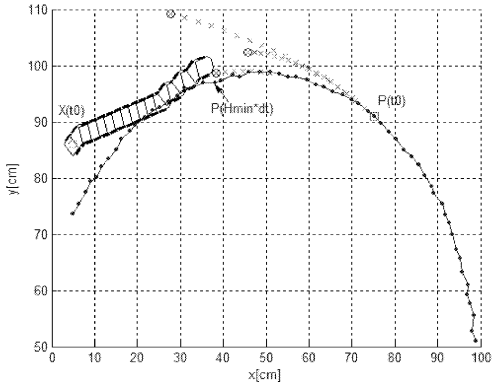
Since the algorithm of control is adopted from the previous work please refer to (Skrzypczyk 2006).

SIMULATION RESULTS

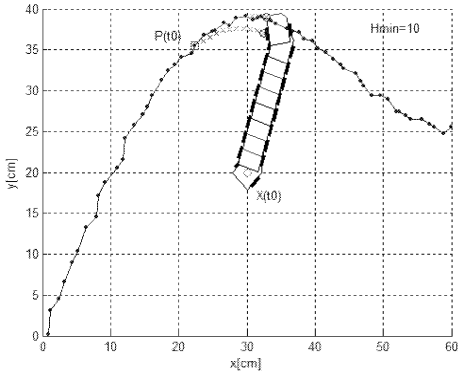
In order to verify the proposed approach a great number of simulations were performed. All the experiments were implemented in MATLAB™ software. In this section results of the two selected experiments are presented. The problem consists in tracking by a robot a target moving along unknown trajectory in the time optimal way. The result of the first simulation is presented in a fig. 3. A plot of a part of the target *P* trajectory is depicted with a dot-solid line. The robot is depicted by a polygon. The process of tracking starts in $t_n=t_{n0}$. Initial state of the robot and the target are equal $X(t_{n0})$=[5 85 45] and $P(t_{n0})$=[75 91] respectively. Successive stages of prediction are depicted with crosses The assumed accuracy of tracking ε is equal ε_T=2 [cm]. The maximal simulated is equal $v_{\max} = 20[cm/s], \omega_{\max} = 340[°/s]$. The accelerations are set to $a_v = 20[cm/s^2], a_\omega = 424[°/s^2]$. The initial speed of the robot is equal to 0 (v_0=0, ω_0=0). In this experiment the algorithm needed 3 corrections to accomplish the task with the set accuracy in the minimal time $t_{\min} = 1.8[s]\,(H_{\min} = 18)$. The second experiment was performed under the same conditions as the first one. Also notation and symbols are the same as before. The target moves along a different trajectory. The initial location of the robot and the target is $X(t_{n0})$=[30 20 45] and $P(t_{n0})$=[22 36]. Also the accuracy of tracking was increased to ε_T=1 [cm]. After the two corrections of the prediction model the robot reached the target in the minimal time $t_{\min} = 1[s]\,(H_{\min} = 10)$. We can observe (fig. 4) effective work of the tracking algorithm. Although the control strategy is based on the prediction model (which can differ much from the original trajectory of the robot) applied method is to the some way robust to the errors of prediction. That results in quite smooth and "reasonable" path of the robot.

CONCLUSION

In the paper an approach to a synthesis of the robust time optimal tracking algorithm is discussed. Presented approach gives very good results what proved simulations that were run. Analyzing obtained results conclusions can be drawn. The algorithm finds the time optimal solution based on the prediction model of the target movement. Therefore the solution obtained may vary from the real time optimal solution with known trajectory of the target. So the notion "time optimality" must be perceived as so called "empirical time optimality". The method gives acceptable results which can be slightly different than the globaly optimal solution. Additional advantages of the proposed method of prediction as well as the tracking method are their simplicity and robustness to uncertain data.



Figures 3: The Result of the First Experiment.



Figures 4: The Result of the Second Experiment

ACKNOWLEDGMENTS

This work has been supported by the founds BK208/Rau1/2006/t.1

REFERENCES

Åström K. and B. Wittenmark 1997. "Computer-Controlled Systems". Prentice-Hall, 1997

Gass, S.L. 2005. "Predictive Fuzzy Logic Controller For Trajectory Tracking of a Mobile Robot". *2005* IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, Helsinki University of Technology, Espoo, Finland, June 28-30, 2005

Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund 2002. "Particle filters for positioning, navigation, and tracking". In *Performance 1983*, A.K. Agrawalla and S.K. Tripathi (Eds.). North-Holland, Amsterdam, 309-323.

Liao, L., Fox, D. Hightower, J., Kautz, H. and D. Schulz. 2003. "Voronoi Tracking: Location Estimation Using Sparse and Noisy Sensor Data". In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-03).

Skrzypczyk K. 2006. "Robust Prediction And Tracking Method For A Mobile Robot Navigation". In Proc. of the ISC Conference, Palermo, Italy pp. 467--471

ON THE USE OF GENERALIZED DERANGEMENTS FOR SCHELLING'S MODEL OF SEGREGATION

Rawan Ghnemat[(1)], Gérard H.E. Duchamp[(2)], Cyrille Bertelle[(1)]

[(1)]LITIS - University of Le Havre, 25 rue Philippe Lebon, BP 540, 76058 Le Havre Cedex, France
email: rawan.ghnemat@litislab.eu, cyrille.bertelle@litislab.eu

[(2)]LIPN - University of Paris XIII, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France
email: ghed@lipn.univ-paris13.fr

KEYWORDS
self-organization, urban modelling, generalized derangement, Schelling's model

ABSTRACT

This paper proposes a definition of Schelling's model of segregation using generalized derangements. Many of urban or territorial modellings are based on decentralized approaches where rule-based systems have to be integrated inside a whole interaction system to describe complex phenomena. The goal of these decentralized modellings is to deal with emergent computing able to detect dynamically emergent organizations in an unsupervized way, thanks to complex systems theory. The convergence of these modern computings is generally hard to study because of the use of asynchronised processes dealing with a number of autonomous entities which are acting and interacting, in non linear way, during the whole simulation. Our approach is to define a non sequential-dependant algorithm, thanks to generalized derangements, and so to use this efficient tool to study some properties on the evolutive process.

INTRODUCTION

Urban or territorial systems are nowadays the topics of decentralized simulations which find roots on two majors advances in the development of scientific researches (Pumain 2006). The first one concerns the development of powerful distributed computer networks which allow to simulate in accurate way, detailed phenomena. The second one which finds major support from the first, is the development of decentralized approaches of modelling. This modern vision of modelling has been mainly enlighted by the complex systems theory (LeMoigne 1999) which allows to deal with the dynamic detection of emergent organizations using swarm optimization processes (Ghnemat et al. 2007), for example. Practical applications deal today with the development of Geographical Information Systems (GIS) which give accurate spatial data and allow with specific platforms (Strout 2006) to link these spatial data with agent-based simulation and emergent computing.
We can observe that humans are more or less sorted into groups of similar people, according sometimes with racial and ethnic features but also in many others ways according to various characteristics. Thomas Schelling proposed in the 1970's a very simple rule-based model able to describe segregation phenomena (Schelling 1971). His description concerns two-dimensional lattice support and it generates from random population, a spatial self-organization which is observed as patterns of clusters of similar persons. The cluster formation does not follow linear dependance from any parameter dealing with the evolution of the rule-based systems. More generally, Schelling model can be used as generic spatial self-organization concept for various kinds of systems: A. Singh and M. Haahr, for example, use a variant of this model to study the topology adaptation in P2P networks (Singh and Haahr 2006).

AN EXACT DEFINITION OF SCHELLING'S MODEL USING GENERALIZED DERANGEMENT

The original Schelling's model can be extended in concurrent version in order to obtain an exact definition of the implementation for each time step, not depending of way the grid is scanned.

Even if the qualitative behavior of Schelling's model is independant of the order of execution od the agents, as long as each agents has the same expected probability to move, the quantitative outcome can be different. Our purpose is to give a non sequential dependant algorithm which mainly allows us to exhibit exact computationn scheme for the whole iterative process.

In the following, we will make use of the (somewhat extended) notion of derangement. For the classical notion see (Stanley 1999):
http://mathworld.wolfram.com/Derangement.html

http://en.wikipedia.org/wiki/Derangement

Here it will be called *generalized derangement.*

Definition 0.1 *Let X, Y be two sets $X \subset Y$. A* generalized derangement *from X to Y is an into (i. e. injective) mapping* [1] $\alpha : X \mapsto Y$ *such that*

$$(\forall x \in X)(\alpha(x) \neq x) \tag{1}$$

To describe Thomas Schelling's concurrent model, we start with a two-dimensional lattice board which is a rectangle of $n \times m$ (n lines and m columns) points (each point will be located by its coordinates (x, y) with $1 \leq x \leq m$; $1 \leq y \leq n$). A *state* of the board will be simply a mapping $s : [1..m] \times [1..n] \mapsto \{0, A, B\}$ [2] indicating whether a point at (x, y) has a value corresponding to

$$\begin{cases} \bullet \text{ nothing } s(x,y) = 0 \\ \bullet \text{ an element of type } A,\ s(x,y) = A \\ \bullet \text{ an element of type } B,\ s(x,y) = B \end{cases} \tag{2}$$

the dynamics of the system will be described by a sequence of states $s_0,\ s_1, \cdots,\ s_n, \cdots$ generated by the following rules.

1. one fixes a threshold (in percent) $0 \leq t \leq 1$;
2. the original state is s_0 (i.e. a distribution of A, B and empty cells along the board);
3. at each step, for each (filled) cell of type X at (x, y), one counts the ratio $\rho(x, y)$ of neighbours of type X over the number of neighbours;
4. if $\rho \geq t$ the cell is marked r (remain), if not it is marked m (move);
5. let M be the set of cells marked "move" and E the set of empty cells;
6. choose randomly (uniform distribution) α among the generalized derangements $M \mapsto M \cup E$;
7. then $s_{n+1}(x, y) = s_n(x, y)$ if the cell was marked r and $s_{n+1}(x, y) = \alpha(s_n(x, y))$ otherwise.

This algorithm is typically controlled by grid diffusion processus using elementary rule-based systems. The rules leading to each move is based on local computation of neighbourhood, however the moves are global, holistic self-organizational. The following section proposes more sophisticated behavior modelled by agent systems.

[1] see `http:en.wikipedia.orgwikiInjective_function`

[2] We can extend the model using greater alphabet cardinal

RANDOM GENERATION FOR THE STUDY OF THE PROPERTIES OF THE DYNAMIC PROCESS

In order to make experiences for the description of emerging phenomena varying the threshold. We have to be able to draw "at random" a generalized derangement $M \mapsto M \cup E$ (step 6). This can be done very efficiently using a step by step random generation of the images. But one must be aware that, given a fixed element of M, say m_0 the set of generalized derangements α such that $\alpha(m_0) \in E$ has not the same cardinality that the set of generalized derangements α such that $\alpha(m_0) \in M$. More precisely, if one denotes $gd(M, E)$ the set of generalized derangements $M \mapsto M \cup E$ and $gd(m, e)$ its cardinality ($m = |M|$ and $e = |E|$), one has

- $gd(0, e) = 1$
- $gd(m, 0) = d_m$, the number of classical derangements, given by the formula

$$\sum_{n=0}^{\infty} d_n \frac{z^n}{n!} = exp(\sum_{n=2}^{\infty} \frac{z^n}{n!}) \tag{3}$$

- $gd(m, e + 1) = gd(m, e) + m\ gd(m - 1, e + 1)$

Moreover, the graphs of generalized derangements are of "exponential class" (Duchamp et al. 2003, Stanley 1999, Wilf 1994) and therefore, the multivariate stastistics of them can be treated by the exponential formula (a formula invented by the Physicist G. E. Uhlenbeck, see (Goulden and Jackson 1983)). One can then harness elegantly the random generation by computing explicitely the branching quantities. The combinatorial study of these probabilities does not meet the purpose of this contribution and will be the subject of a forthcoming paper.

CONCLUSION AND PERSPECTIVES

This paper presents a definition of Schelling's model using generalized derangements. Thanks to the non sequential dependant algorithm formalism proposed here, we gives an efficient tool to study the properties of the system dynamic. Combinatorial studies and implementation of this segregation model description are in progress. Convergence properties could be then studied and are expected to be obtained.

REFERENCES

Benenson I. and Torrens P.M., 2004. *Geosimulation.* Wiley.

Daudé E. and Langlois P., 2006. *Modélisation et simulation multi-agents*, Hermès Sciences, Lavoisier, chap. Comparaison de trois implémentations du modèle de Schelling. 411–442.

Duchamp G.; Solomon A.; Penson K.; Horzela A.; and lasiak P.B., 2003. *One-parameter groups and combinatorial physics*. In M.N.H. J. Govaerts and A.Z.M. (eds.) (Eds.), *Proceedings of the Symposium Third International Workshop on Contemporary Problems in Mathematical Physics (COPROMAPH3)*. World Scientific Publishing 2004, Port-Nove, Benin, 436. ArXiv: `quant-ph/04011262`.

Ghnemat R.; Bertelle C.; and Duchamp G.H., 2007. *Community Swarm Optimization*. In *workshop on Emergent Properties for Natural and Artificial Complex Systems, ECCS'07*. Dresden, Germany.

Goulden I.P. and Jackson D.M., 1983. *Combinatorial Enumeration*. John Wiley & Sons Inc.

Laurie A.J. and Jaggi N.K., 2003. *Role of 'Vision' in neighborhood racial segregation: a variant of the Schelling segregation model*. *Urban Studies*, 40(14), 2687–2704.

LeMoigne J.L., 1999. *La modélisation des systèmes complexes*. Dunod.

Omer I., 2005. *How ethnicity influences residential distributions: an agent-based simulation*. *Environment and Planning B: Planning and Design*, 32, 657–672.

Pumain D. (Ed.), 2006. *Hierarchy in Natural and Social Sciences*. Springer.

Schelling T.C., 1971. *Dynamic Models of Segregation*. *Journal of Mathematical Sociology*, 1, 143–186.

Singh A. and Haahr M., 2006. *Creating an adaptive network of hubs using Schelling's model*. *Communication of the ACM*, 49(3), 69–73.

Stanley R., 1999. *Enumerative combinatorics*. Cambridge University Press.

Strout N., 2006. *What is Agent Analyst*. In *The 2006 ESRI International User Conference*. San Diego.

Weiss G. (Ed.), 1999. *Multiagent Systems*. MIT Press.

Wilf H., 1994. *Generating Functionology*. Academic Press, New York.

EMOTION: APPRAISAL-COPING MODEL FOR THE "CASCADES" PROBLEM

Karim Mahboub[(1)], Evelyne Clément[(2)], Cyrille Bertelle[(1)] & Véronique Jay[(1)]

[(1)]LITIS Laboratory, University of Le Havre
25, rue Philippe Lebon, BP 540 76058 Le Havre Cedex, France
Karim.Mahboub@litislab.eu
Cyrille.Bertelle@litislab.eu
Veronique.Jay@litislab.eu

[(2)]Psy.CO Laboratory, University of Rouen
Rue Lavoisier 76821 Mont-Saint-Aignan, France
Evelyne.Clement@univ-rouen.fr

ABSTRACT

Modelling emotion has become a challenge nowadays. Therefore, several models have been produced in order to express human emotional activity. However, only a few of them are currently able to express the close relationship existing between emotion and cognition. An appraisal-coping model is presented here, with the aim to simulate the emotional impact caused by the evaluation of a particular situation (appraisal), along with the consequent cognitive reaction intended to face the situation (coping). This model is applied to the "Cascades" problem, a small arithmetical exercice designed for ten-year-old pupils. The goal is to create a model corresponding to a child's behaviour when solving the problem using his own strategies.
Keywords: emotion modelling, decision making, appraisal-coping model.

INTRODUCTION

As the study of emotion is becoming crucial today, in several fields of study such as neurology or psychology, computer science is getting more and more involved in the process of finding new models for representing emotions. Since the middle of the 19th century, psychologists, biologists, but also neurologists have tried to produce models designed to unravel the emotional processes. Scientists like Bechara and Damasio [3] have even proved that human emotional activity has an indispensable impact on decision making.

The aim of this article is to globally draw the picture of the latest models of emotion in computer science, starting from a psychologist point of view. The appraisal-coping model will be presented more precisely, as well as an existing application example. Afterwards, we will present a small toy problem illustrating a decision making problem, using the appraisal-coping strategy. Eventually, a small conclusion will introduce future aspects to be developped and other prospects.

STATE OF THE ART

From the most recent approaches of emotion modelling, two major cognitive types of research models have been developed: hierarchical and componential models (see Baudic & Duchamp, 2006 [1]).

According to the hierarchical approach, emotions have their origin from early stages of development. Indeed, emotions are hierarchically organised with numerous discrete emotions at a basic level and emotional dimensions at a higher level. Fundamental emotions like thirst or fear are elements intended to build more sophisticated emotions, ending up with very complex feelings such as jealousy or pride [7]. Emotions at the basic level have an important adaptive function and are directly linked to the body stimuli and effectors.

In componential models, emotions have qualitatively different facets [23]. The so-called "emotional response triad" is composed of the three main components for the emotion production: subjective experience, peripheral physiological responses and motor expression, to which some theorists include two other components, cognitive and motivational. The componential approach deals with the relative role assigned to each of these components. Then, emotions are created by stepping through all parts of the process, from the cognitive perception, until the actual response. Lazarus [13] and Scherer [20] are usually associated with this approach.

APPRAISAL THEORIES

Appraisal theories suggest that emotion is the result of underlying mechanisms including the subjective evaluation of the significance of a situation and its organism circumstances (appraisal), and the coping mechanisms that guide and provide adaptive responses (Frijda, 1986 [9]; Lazarus, 1991 [14]; Scherer, 1984 [20]; Scherer,

Schorr & Johnstone, 2001 [24]; Smith & Lazarus, 1990 [26], 1993 [27]). As noted by Gratch and Marsella [10]: "Appraisal theories posit that events do not have significance in of themselves, but only by virtue of their interpretation in the context of an individual's beliefs, desires, intentions and abilities" (Gratch & Marsella, 2004, p. 273). The significance of an event is supposed to be evaluated on a number of criteria such as its relevance for one's well-being, its conduciveness for one's plans and goals, and the ability to cope with such consequences.

In the framework of the Scherer's Component Process Model (Scherer 1984 [20], 2001 [24]), Sander, Grandjean & Scherer (2005 [19]) describe emotion "as an episode of interrelated, synchronized changes in the states of all or most of the five organismic subsystems[1] in response to the evaluation of an external or internal stimulus event as relevant to major concerns of the organism.", (p.318). From this point of view, rather than static and basic states of the organism (e.g. Ekman, 1984 [8]; Izard, 1971 [11]), emotions are a dynamic process whose components are the cognitive component which function is the evaluation of objects and events, the peripheral efference component which regulates the system, the motivational component which prepares and guides the actions, the motor expression component which steadies communication of reaction and behavioural intention, and the subjective feeling component which monitors the internal state and environment interaction. In other respects, this model postulates that changes in one subsystem will tend to elicit related changes in other subsystems.

APPRAISAL-COPING EXAMPLE

Based on the appraisal-coping approach, several new models have been conceived (see figure 1). Gratch & Marsella [10] have produced a domain independent model using cognitive maps. This model is intended to manipulate appraisal variables to analyse the present and past situations, and to design the future decisions to make and the coping strategies to adopt. Generally speaking, the appraisal-coping approach offers a very precise model of cognitive and emotional processes in decision making (previous works have been carried out on this subject, see [16]).

On the one hand, the environment is appraised with respect to one's goals and beliefs. This evaluation is realised through a certain number of variables defining the different appraising dimensions to be taken into account, such as the relevance (*Does the event require attention or adaptive reaction?*) or unexpectedness (*Was the event predicted from past knowledge?*).

On the other hand, one has to cope with the situation appraised before by using coping strategies. These different strategies offer a great range of possibilities, from the perfect control of the situation until the total resignation, when facing the problem. The coping outcome alters the person-environment relationship not only by modifying the environment itself related to the problem, but also by changing the interpretation and willing through emotional aspects.

In order to represent the information about the situation, Gratch & Marsella [10] use causal maps. In the following scenario (figure 2), an oncologist, Dr. Tom, is supposed to help an eleven-year-old boy, Jimmy, for his stage 4 inoperable cancer, either by giving him morphine (which relieves the pain but hastens death) or leaving him suffer (and letting him prolong his life). A causal map (see figure 2) represents the past and present situations, as well as the possible decisions to make and their expected consequences.

This causal map can evolve with the modification of the parameters. For example, U_{Jim} is how Dr. Tom appreciates Jim's desirability for the corresponding event. This value can be updated after the coping process, dealing with emotional aspects. Once Dr. Tom re-appraises the consequences of his decision (by decreasing the probability parameter of the "Death hastened" event), he accepts to give Jim morphine, following Jim's mother request.

However, the major problem when using this technique is the storage of the information taken from the environment. In fact, we have two options.

The first option is to put all the information we need explicitly on the object to be evaluated. For instance, if an individual is watching a photograph of his wife and kids, the variables indicating pleasantness or pride are to be described on the photograph. With this method, each object in the environment is clearly identified as pleasant or mysterious or annoying, etc. The implementation is therefore easier, and the possibility of interacting with the emotional representation of the object is hence trivial. The main problem is a lack of flexibility, especially if more than one individual has to evaluate the same object.

The second option is the internal storage of the objects data, directly into the brain, using a memory strategy (Tulving et al., 1972 [29]). With this technique, memory is divided into several categories, each of which stores different kind of information. On the one hand, the long-term memory system is composed of the episodic memory (i.e. it refers to knowledge of episodes and facts that can be consciously recalled and related) and the semantic memory (underlying absolute knowledge and language; semantic memory is context-independent). On the other hand, the short-term memory, also known as working memory stores the current context-related data. For each of the above options, emotion is supposed to

[1]Organismic subsystem (and their major substata are the following): Information processing (Central Nervous System – CNS), Support (CNS, Neuro-Endocrine System, Autonomic Nervous System), Executive (CNS), Action (Somatic Nervous System), Monitor (CNS), from Sander, Grandjean & Scherer (2005, [19]).

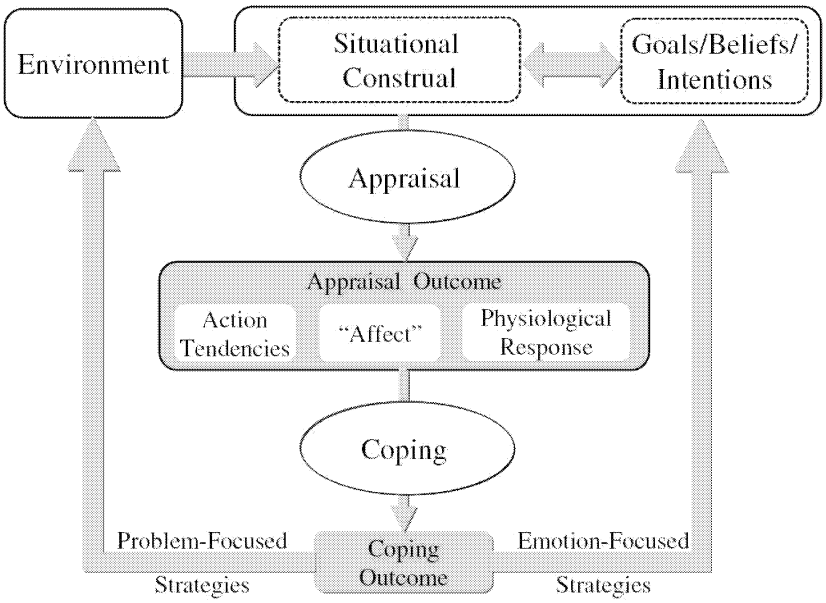


Figure 1: The cognitive-motivational-emotive system. Adapted from Smith an Lazarus (1990, [26]).

be triggered from the appraisal processes. Indeed, the evaluation activity requires knowledge, and emotion is part of the memory processes of encoding, storage and retrieval (Tulving & Thomson, 1973 [30]). For modelling purpose, we assume that memory is split into three categories :

- The semantic or factual memory which stores the global knowledge of the world and the information considered to be facts, like *"Paris is the capital of France"*.
- The episodic or autobiographical memory which contains the personal events that happened in an individual's life. This type of memory is strongly linked with a spatio-temporal context.
- The working memory is the current dynamic representation that an individual has in mind when solving a particular problem.

Each time a memory element is encoded (i.e. added into the brain), it is stored along with the current emotional learning context e_{t_0}. Later on, when the information is required and must be found in the memory, the current emotional context e_t will be compared to the former emotional learning context related to the required memory item. If the old emotional context equals the current one (i.e. if $e_{t_0} = e_t$) the retrieval mechanism will be facilitated. On the contrary, if the two values are different ($e_{t_0} \neq e_t$), the retrieval process will be made more difficult.

THE "CASCADES" PROBLEM

The "Cascades" problem is a puzzle-like situation in which the goal is to fill up the grid with numbers according to the following instruction: "Each box contains the sum of the numbers situated above it. Look for the missing numbers in the grid".
The initial state and the first solving step are presented hereafter (figure 3).
Ten-year-old children will be invited to solve the problem. In order to study how the emotions are elicited over the course of the problem, emotional manifestations will be recorded without interruption during the problem-solving activity. In line with the work of Clément & Duvallet (2007 [6]), we will focus on two kinds of response components: the physiological – electrodermal activity – and the expressive – facial expressions – components.
Although the definition of emotions remains controversial, some researchers distinguish emotions and other related notions as mood or personality traits on the basis of their behavioural time course and intensity: emotions are defined as short-lived behavioural dispositions, moods are of longer duration and lower intensity, while personality traits reflect relatively stable behavioural tendencies.
Nevertheless, it is generally assumed that emotions may be evaluated by three kinds of responses: the physiological responses which the electrodermal activity and the heart rate are the most widely used, the expressive responses including facial, vocal, gestural and postural expressions and the subjective responses based in part on verbal report (see Bauer, 1998 [2]; Boehner, DePaula,

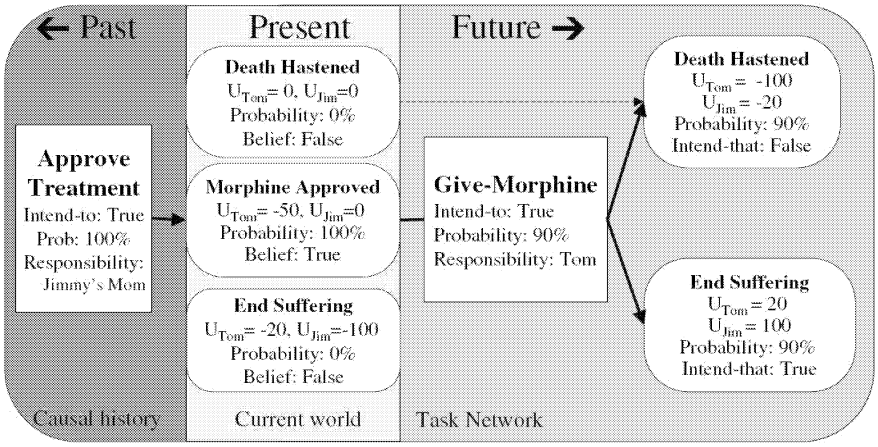


Figure 2: Dr. Tom's causal interpretation at the end of the scenario.

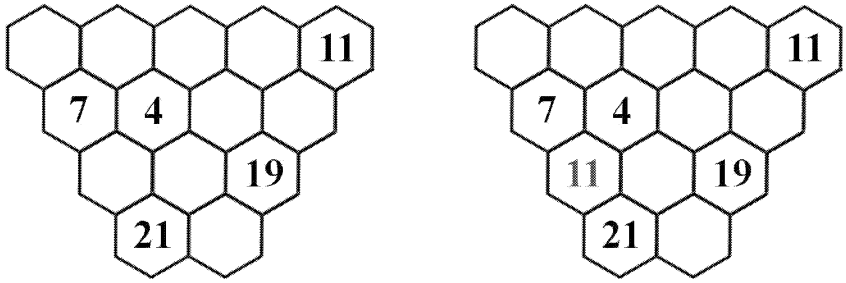


Figure 3: The "*Cascades*" initial grid and the first solving step.

Dourish, & Sengers, 2007 [4], for critical reviews).

Concerning the physiological component, and in particular the electrodermal activity, the study of Pecchinenda and Smith (1996) provides psychological significance of spontaneous skin conductance activity. In that work, participants were given to solve a set of anagrams which difficulty was manipulated by both the objective difficulty (easy, moderately, difficult, extremely difficult) and the amount of time available to solve the problems (30 vs. 120 seconds). The authors demonstrate that the skin conductance activity during problem solving is correlated to the appraisals of coping potential: in a difficult problem, appraisals of coping potential based on self-report are especially low and produce selective disengagement of the task, yielding reduced skin conductance activity. The spontaneous electrodermal activity is interpreted as reflecting task engagement (Pecchinenda, 2001 [17]).

Moreover, the results of Clément and Duvallet (2007 [6]) support the idea that skin conductance activity is a convergent measure of appraisal-related processes and that facial expressions reflect the appraisals of the events according to their conduciveness for the goal (Kaiser & Wehrle, 2001 [12]; Scherer, 1999 [21]; Smith, 1991 [25]; Smith & Scott, 1997 [28]).

THE APPRAISAL-COPING MODEL

The appraisal-coping model adaptation to the "Cascades" problem (see figure 4) allows us to analyse in a more accurate way the children's cognitive emotional activities. The appraisal step deals with the evaluation and the prediction of the plan selected by the child in order to solve the problem. According to this appraisal step, the child will fill in a hexagon, following his selected plan, or correct a previous result, or if the appraisal step leads to a bad evaluation, he will change his plan (we observe that the exercice instructions are interpreted in many different ways, especially for children with school difficulties).

The coping strategy is the actual decision a child will take with respect to his previous choices. The coping process is usually accompanied by an emotional reaction which depends on the appraisal consequences. For instance, if the child decides to reinforce his strategy, his emotional state will be positive. On the contrary, if he continually changes his plans, the situation will end up with a progressive disengagement for the task.

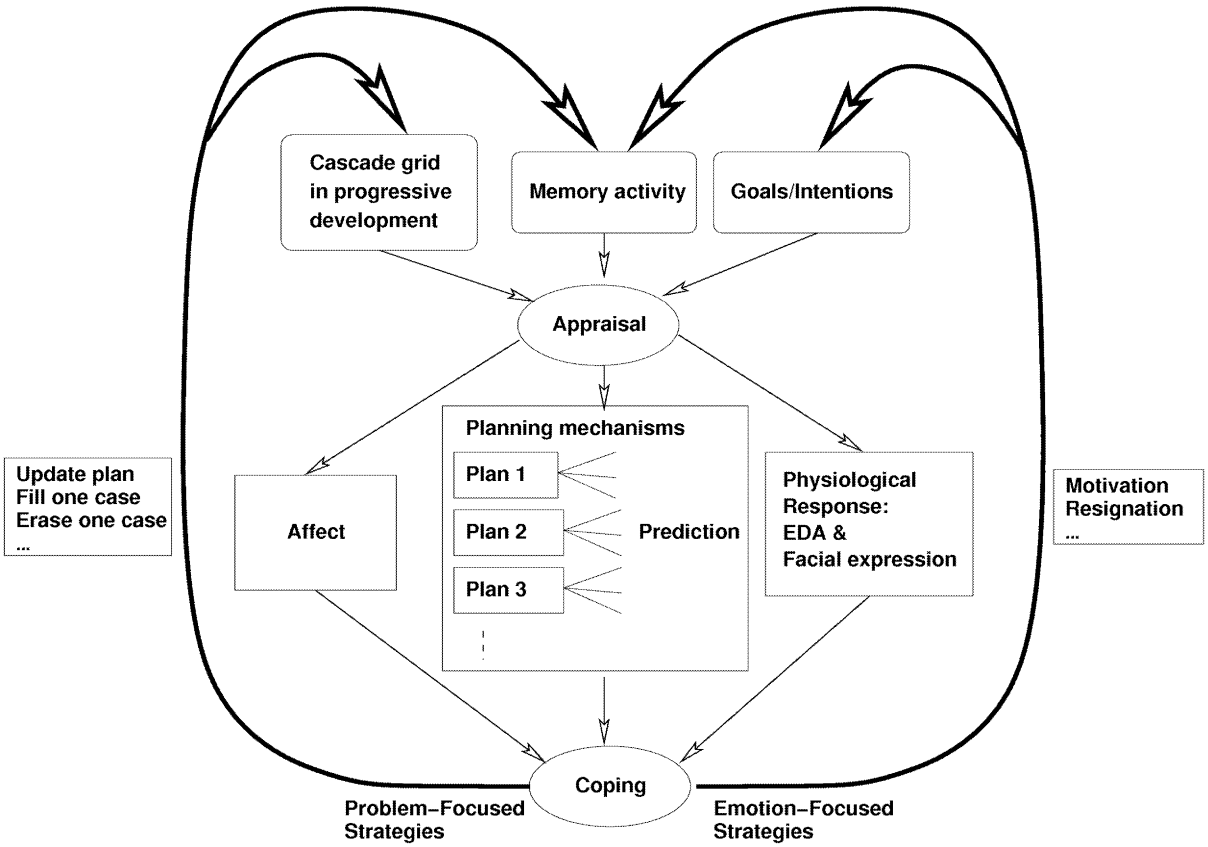


Figure 4: The appraisal-coping model applied to the "Cascades" problem.

CONCLUSION AND PROSPECTS

Contrary to the classical models which study cognition in a "cold" way, independently from any emotional process, we are now trying to unravel the mysteries underlying the emotion-cognition interaction. This new challenge makes researchers produce a new generation of cognitive models, based on more accurate systems, such as the appraisal-coping approach.

The multidisciplinary work carried out so far allows us a better understanding of emotion mechanisms, by bringing out two complementary approaches:

- The produced model aims to analyse the experimental data.
- The experimental data come to strengthen or question the existing model.

As a future work, the strategies selected in the coping process are to be examined more precisely, in order to extract cognitive-emotional individual profiles, and especially when it comes to help pupils with school difficulties.

REFERENCES

[1] S. Baudic and G.H.E. Duchamp. Emotions: theoretical models and clinical implications. In *European Simulation and Modelling Conference '2006*, 2006.

[2] R.M. Bauer. Physiologic measures of emotion. *Journal of Clinical Neurophysiology*, 15:388–396, 1998.

[3] A. Bechara, H. Damasio, and A.R. Damasio. Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex*, 10:295–307, 2000. `http://cercor.oxfordjournals.org/cgi/content/full/10/3/295`.

[4] K. Boehner, R. De Paula, P. Dourish, and P Sengers. How emotion is made and measured. *International Journal of Human-Computer Studies*, 65:275–291, 2007.

[5] E. Clément. La flexibilité représentationnelle. *Mémoire de HDR - Université de Rouen*, 2007.

[6] E. Clément and D. Duvallet. Emotions elicited by impasses and subgoal achievements in problem-solving. In *CEM 07, Cognition, Emotion and*

Motivation International Congress, Hammamet, Tunisia, 2007.

[7] D. Denton. *The Primordial Emotions: The Dawning of Consciousness*. Oxford University Press, USA, 2006.

[8] P. Ekman. Expression and the nature of emotion. In K.R. Scherer and P. Ekman, editors, *Approaches to emotion*. Erlbaum, Hillsdale, NJ, 1984.

[9] N. Frijda. *The Emotions*. Cambridge University Press, Cambridge, 1986.

[10] J. Gratch and S. Marsella. A domain-independant framework for modeling emotion. *Cognitive Systems Research*, 5:269–306, 2004.

[11] C.E. Izard. *The face of emotion*. Appleton (Century/Crofts), New York, 1977.

[12] S. Kaiser and T. Wehrle. Facial expressions as indicators of appraisal processes. In K.R. Scherer, A. Schorr, and T. Johnstone, editors, *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, New York, 2001.

[13] R.S. Lazarus. Thoughts on relations between emotion and cognition. *The American Psychologist*, 37:1014–1019, 1982.

[14] R.S. Lazarus. *Emotion and Adaptation*. Oxford University Press, New York, 1991.

[15] K. Mahboub. Modélisation des processus émotionnels dans la prise de décision. Master's thesis, Université du Havre, France, 2006.

[16] K. Mahboub. Simulation of emotional processes in decision making. In *European Simulation and Modelling Conference '2006*, 2006.

[17] A. Pecchinenda. The psychophysiology of appraisals. In K.R. Scherer, A. Schorr, and T. Johnstone, editors, *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, New York, 2001.

[18] A. Pecchinenda and C.A. Smith. The affective significance of skin conductance activity during a difficult problem-solving task. *Cognition and Emotion*, 10:481–503, 1996.

[19] D. Sander, D. Grandjean, and K.R. Scherer. A systems approach to appraisal mechanisms in emotion. *Neural Networks*, 18:317–352, 2005.

[20] K.R. Scherer. On the nature and function of emotion. a component process. In K.R. Scherer and P. Ekman, editors, *Approaches to Emotion*, pages 293–317. Erlbaum, Hillsdale, NJ, 1984.

[21] K.R. Scherer. Appraisal theories. In T. Dalgeish and M. Power, editors, *Handbook of cognition and emotion*. Wiley, Chichester, 1999.

[22] K.R. Scherer. Appraisal considered as a process of multi-level sequential checking. In K.R. Scherer, A. Schorr, and T. Johnstone, editors, *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, New York, 2001.

[23] K.R. Scherer and M. Peper. Psychological theories of emotion and neuropsychology research. In F. Boller and J. Grafman, editors, *Handbook of Neuropsychology*, volume 5: Emotional behavior and its disorders, pages 17–48. Elsevier, Amsterdam, 2001.

[24] K.R. Scherer, A. Schorr, and T. Johnstone. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, New York, 2001.

[25] C.A. Smith. The self, appraisal, and coping. In C.R. Snyder and D.R. Forsyth, editors, *Handbook of social and clinical psychology: The health perspective*. Pergamon, New York, 1991.

[26] C.A. Smith and R.S. Lazarus. Emotion and adaptation. In L.A. Pervin, editor, *Handbook of personality: Theory and research*, pages 609–637. Guilford, New York, 1990.

[27] C.A. Smith and R.S. Lazarus. Appraisal components, core relational themes, and the emotions. *Cognition and emotion*, 7:233–269, 1993.

[28] C.A. Smith and H.S. Scott. A componential approach to the meaning of facial expressions. In J. Russel and J.M. Fernandez-Dols, editors, *The psychology of facial expression*. Cambridge University Press, Cambridge, 1997.

[29] E. Tulving. Episodic and semantic memory. In E. Tulving and W. Donaldson, editors, *Organisation of memory*, pages 381–403. Academic Press, New York, 1972.

[30] E. Tulving and D.M. Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, 80:352–373, 1973.

ON THE USE OF ANT COLONY IN ASSOCIATIVE CLASSIFICATION

Kifaya Qaddoum & Fadi Thabtah
Philadelphia University
Amman, Jordan
E-mail: sweetkq@hotmail.com,
ffayez@philadelphia.edu.jo

Cyrille Bertelle
LITIS – University of Le Havre
25 rue Philippe Lebon, BP 540,
76058 Le Havre Cedex, France
E-mail: cyrille.bertelle@litislab.eu

KEYWORDS
Ant Colony, associative classification, data mining, classification rules

ABSTRACT

So far, ant algorithms have showed magnificent research interest within the search/optimisation community. In Data Mining search space, Associative Classification [3] (AC) approaches proved to be strongly able to extract more accurate classifiers than traditional classification techniques, In this paper the two are brought together. An investigation of the ant colony algorithms to present a variant of them for associative classification mining. In the proposed algorithm, each class is labeled and assigned to represent the consequent part of the rules within a processor. The ants groups are attached to the main processor to search for the antecedent part of the rules. The ants, depending on the pheromone, and heuristic information, select the values of the attributes according to the importance of an attribute to the class. The proposed algorithm is expected to discover classification rules with magnificent enhancement on accuracy and less redundancy than comparing methods.

INTRODUCTION

Building effective classification systems is one of the main tasks of data mining and machine learning. Past researches have produced many techniques (e.g. Decision Tree [12], Naive-Bayes [10], Support Vector Machines [5], Neural Networks [13], and Statistical approaches [26], [8]. In recent years a new classification approach that integrates classification with association rule mining called associative classification arises, which is considered a promising technique that produces highly accurate classifiers, AC in many experiments proved to give more accurate rules, and less error rate relating to the produced rules, comparing with other leading techniques in Data Mining Classification .

AN EXACT DEFINITION OF ASSOCIATIVE CLASSIFICATION

Classification rule mining and association rule mining are two important data mining techniques [4]. The target of mining is not pre-determined for association rule mining, while for classification rule mining there is one and only one pre-determined target, i.e., the class.

Associative classification [3] proposed the integration of association rule mining and classification. Association rule is unsupervised learning that describes the co-occurrence among data items in a large amount of collected data [1]. Whereas, associative classification is a supervised task that predicts the class label of test cases.

This section introduces the associative classification technique. A training dataset R is characterized by a schema (A1. . . Ak, c), where (A1, . . . , Ak) are k distinct attributes and c is a class attribute. Each tuple in R is a data object, which is associated to a unique identifier called tid. The attributes may have either a categorical or a continuous domain. For categorical attributes, all values in the domain are mapped to consecutive positive integers. For continuous attributes, the value range is discretized into intervals, which are mapped into consecutive positive integers.1 In this way, all attributes are treated uniformly.

Each data object in R can be described as a collection of pairs (attribute, integer value), plus a class label (a value belonging to the domain of the class attribute c). Each pair (attribute, integer value) is called item in the remaining part of the paper.

In associative classification, classification rules are used to model the most relevant properties characterizing classes of data, and to predict the class label for unknown (unlabeled) data. A classifier is a function from 2^I to C that allows the assignment of a class label to a data object. A classifier, able to predict the class label for data objects with high accuracy, is generated from a collection of transactions (i.e., data objects with a label, also called training dataset). In associative classification, the classifier is generated by selecting the most appropriate set of association rules. A rule $r : \mathcal{X} \rightarrow c$ classifies or matches a data object d when $\mathcal{X} \subseteq$ d. In this case, rule r assigns class label c to data object d.

Several measures have been proposed to quantify the "interestingness" or the quality of an association (and a classification) rule. Frequently used quality indices are support and confidence [1]. In our setting, these measures correspond to the support and confidence of the l-itemset

encoding the classification rule. For an arbitrary classification rule $\mathcal{X}c$, the support sup($\mathcal{X}c$) is the fraction of transactions in D which contain $\mathcal{X}$ and are labeled by class label c. For the rule antecedent $\mathcal{X}$, the support sup($\mathcal{X}$) is the fraction of transactions in D which contain $\mathcal{X}$.

ANT COLONY ALGORITHM FOR CLASSIFICATION RULES DISCOVERING

Ant colony techniques have been successfully applied to many problems including the Travelling Salesman Problem [8], Quadratic Assignment Problem [8], Job-Shop Scheduling Problems [9], generic constraint satisfaction problems [10], University Course Timetabling Problems [11], cutting and packing [2] graph colouring problems [9] and load balancing in simulation distribution [15]. The standard technique is a constructive one: a colony of ants moves in a solutions space and evolve in collaborative way to find the best solution. Each ant contributes to this collaborative construction by making decisions stochastically, using existing problem constraints and heuristics combined with experience (which is analogous to a substance called pheromone). The colony then reinforces decisions in the construction process according to their successes by adding pheromone, which also decays to mitigate against poorer decisions. Our purpose in this paper is to investigate the ant algorithm technique as a means of constructing effective class labels associated with an attribute value in the training data set rather than just the single most obvious class.

For our purpose, we define a weighted graph G = (V,E) such that V is the set of vertices corresponding to the class labels and E is the set of edge (u,v) represent the transactions between two nodes u and v. The edge weight value corresponds to w(u,v) is sup(u), the fraction of transactions which contain u. At each step, a ant will choose to move, crossing an edge (u,v), according to a probabilistic value p(u,v):

$$p(u,v) = (w(u,v))^{\alpha} (c(u,v))^{\beta} / (\Sigma_{w \in \gamma u} (w(u,w))^{\alpha} (c(u,w))^{\beta}) \quad (1)$$

Where γ_u is all the state reachable from u, α and β are arbitrary parameters and c(u,v) is the pheromone rate deposite by the ant colony on the edge (u,v). When a ant crosses an edge, it deposite a pheromone value on it. After a complete iteration, the whole pheromone rate on each edge (u,v) is computed by

$$c^{t+1}(u,v) = \Sigma_{k \in A} c^{t+1}(u,v,k) + \rho\, c^{t}(u,v) \quad (2)$$

where $c^{t+1}(u,v,k)$ is the pheromone value deposite by the ant k from the set A, at the iteration t+1, ρ is an evaporation factor which allow to decrease the previous pheromone rate. This negative feed-back allow to forget some initial and non optimal solution to make emerge new ones in dynamic way.

This distributed computation contribute to find appropriate transations path for emergent association classification.

REFERENCES

[1] Agrawal, R. and Srikant, R. 1994. "Fast algorithms for mining association rules." VLDB-94, 1994.

[2] E. Burke and G. Kendall, Applying Ant Algorithms and the No Fit Polygon to the Nesting Problem, Proceedings of 12th Australian Joint Conference on Artificial Intelligence, Sydney, Australia, December 6-10, 1999, Lecture Notes in Artificial Intelligence vol. 1747, Foo, N. (Ed), pp. 453-464, 1999.

[3] Baralis, E. And Garza, P. 2002. A lazy approach to pruning classification rules. In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02). IEEE Computer Society Press, Los Alamitos, CA, 35–42.

[4] .B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In Proc. of 4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD), Aug. 1998.

[5] Cristianini, N. And Shawe-Taylor, J. 2000. An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, U.K.

[6] D. Costa and A. Hertz, Ants can colour graphs, Journal of the Operations Research Society 48, pp. 295-305, 1997.

[7] Duda, R. And Hart, P. 1973. Pattern Classification and Scene Analysis. John Wiley and Sons, New York, NY.

[8] M. Dorigo, V. Maniezzo and A. Colorni, The Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man and Cybernetics Part B, vol. 26, no. 1, 1996, pp. 1-13.

[9] M. Dorigo, V. Maniezzo and A. Colorni. Distributed optimization by ant colonies Proceedings of the first European Conference on Artificial Life Paris, ed. By FJ Varela and P Bourgine, MIT/Press/Bradford Books, Cambridge, Massachusetts: pp. 134-142, 1992.

[10] S. Pimont and C. Solnon, A Generic Ant Algorithm for Solving Constraint Satisfaction Problems, Abstract Proceedings of ANTS' 2000 From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, pp. 100-108, 2000.

[11] K. Socha, J. Knowles and M. Sampels, A MAX-MIN Ant System for the University Course Timetabling Problem, Proceedings of the Third International Workshop on Ant Algorithms (ANTS'02), Springer LNCS vol. 2463, pp. 1-13, 2002.

[12] Quinlan. J. C4.5: Programs for machine learning. Morgan Kaufmann, 1993.

[13] R. Lippmann. An introduction to computing with neural nets. IEEE ASSP Magazine, 4(22), 1987.

[14] Thabtah, F, Cowling, P and Peng, Y, 2004, MMAC: A new multi-class, multi-label associative classification approach. In Proceedings of the 4th IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, pp. 217–224.

[15] Bertelle C, Dutot A, Guinand F and Olivier D, 2007, "Organization detection for dynamic load balancing in individual-based simulation", Multiagent and Grid System, pp 141-163, Volume 3, Number 1, 2007.

ARTIFICIAL INTELLIGENCE NEURAL NETS AND FUZZY SYSTEMS

ARTIFICIAL INTELLIGENCE

LINEAR AND INTEGER PROGRAMMING LARGE SCALE HEURISTIC FOR STRIPS PLANNING

Adam Galuszka
Institute of Automatic Control
Silesian University of Technology
44-100 Gliwice, Poland
E-mail: Adam.Galuszka@polsl.pl

KEYWORDS
Block world, STRIPS system, large scale problems, linear programming, computational efficiency.

ABSTRACT

STRIPS planning is a difficult computational problem. One way to increase efficiency of searching for a solution may be a transformation of a problem to another problem and then search for a solution of the transformed problem. In this work a simple STRIPS problem has been transformed to two problems: linear and integer programming. The transformation from planning to Linear Programming is based on mapping of conditions and operators in each plan step to variables. Truth-values of conditions are mapped to 0 and 1 values. For linear programming the objective function reaches the maximum if the goal situation is true in last step of planning. The efficiency and quality of this approach has been analyzed and compared.

INTRODUCTION

Planning belongs to fundamental AI domains. Examples of planning applications are manufacturing, production planning, logistics and agentics.

Planning problem is a problem of finding a set of actions (often also called operators) which transform an initial state into a desired goal situation. It should be distinguished from scheduling – well-known and frequently used technique of improving the cost of a plan. Planning is understood as causal relations between actions, while scheduling is concerned with metric constraints on actions (Backstrom 1998). When all states of a planning problem (including an initial and a goal) are described by a given set of predicates, then the problem is called STRIPS planning problem (Nilson 1980). There are many applications of the planning problems in industrial processes, production planning, logistics and robotics (Backstrom 1998). The STRIPS system has been successfully applied in planning modules of Deep Space One Spacecraft (Weld 1999) and for elevators control in Rockefeler Center in New York (Koehler and Schuster 2000). In these applications it is shown that planning algorithm in practice has to be efficient in time and generate high quality plans (close to optimal).

In this case, the planning problem environment is modelled as a Block World using the STRIPS representation. This domain is often used to model planning problems (Slaney and Thiebaux 2001, Boutilier and Brafman 2001, Kraus et al. 1998, Smith and Weld 1998, Galuszka and Swierniak 2005) because of complex action definition and simple physical interpretation. One can say that starting from 1970s, STRIPS formalism (Nilson 1980) seems to be very popular for planning problems (Weld 1999).

Generally, the problem is that planning problems are very hard combinatorial problems. This hardness is especially characteristic for domain-independent algorithms (Ambite and Knoblock 2001) and it corresponds to difficulties with constructing general solvers. However, it should be noted that even for methods specific for certain domains (e.g. for block world), planning problems usually remain difficult (Bylander 1994). The complexity of planning problems strongly depends on the complexity of the actions defined for the assumed domain (also Bylander 1994).

Motivation

In general, planning with complete information is PSPACE-complete problem (Bylander 1994). In Block World the problem of optimal planning is NP-complete (also Bylander 1994). Planning in the presence of incompleteness is much more complicated (Blythe 1999) and belongs to the next level in the hierarchy of completeness. Precisely speaking, if the incompleteness is modelled by a set of possible initial states, then planning problem is NP^{NP} -complete or $\Sigma_2 P$ – complete (Baral at al. 2000). Because of complexity, in this paper a heuristic of polynomial transformation of STRIPS planning problem to linear programming problem (LP) is used (Bylander 1997). This is done because LP problems are known to be computational easy (Chaczijan 1979). The cost of this approach is that LP result may be non-interpretable for some initial states of STRIPS planning problem (what is followed by assumption $P \neq NP$).

Contribution

The following problems are presented and analyzed:

- polynomial transformation of Block World to Linear Programming Problem,
- proposition of additional heuristic,
- solution using large scale algorithm and binary integer algorithm.

Organization of the paper

The paper is organized as follow: In section 2 the STRIPS language and block world environment is defined. In section 3 the transformation of STRIPS planning to Linear and Integer Programming is described. In section 4 the additional heuristic is proposed. Examplary simulations are shown in section 5. All is concluded in section 6.

STRIPS SYSTEM AND BLOCK WORLD

In general, STRIPS language is represented by four lists (C; O; I; G) (Bylander 1994, Nilson 1980):
- a finite set of ground atomic formulas (C), called conditions;
- a finite set of operators (O);
- a finite set of predicates that denotes initial state (I);
- a finite set of predicates that denotes goal state (G).

The initial state describes the physical configuration of the blocks. This description should be complete i.e. it should deal with every true predicate corresponding to this state. The goal situation describes what should be true. Each goal consists of subgoals and has a form of conjunction of predicates. This description does not need to be complete, i.e. does not need to describe a state of the problem.

The algorithm results in an ordered set of operators which transforms the initial state I into a state with true predicates mentioned in the goal situation G. Operators in STRIPS representation consist of three sublists: a precondition list (*pre(o)*), a delete list (*del(o)*) and an add list (*add(o)*). The precondition list is a set of predicates that must be satisfied to apply this operator. The delete list is a set of predicates that will be false after applying the operator and the add list is a set of predicates that are true after the operator is applied. The two last lists show the effects of applying the operator into a current problem state ($S \subset C$).

Let an operator $o \in O$ takes the form $pre(o) \rightarrow add(o), del(o)$. Following (Koehler and Hoffmann 2000), the set of operators $<o_1, o_2, ..., o_n>$ in a plan is denoted by P^O.

If an operator $o \in O$ is applied to the current state of the problem then the state is modified. This modification is described by function *Result*:

$$Result(S, <o>) = (S \cup add(o)) \setminus del(o) \text{ if } pre(o) \subseteq S, S \text{ in the opposite case} \quad (1)$$

and

$$Result(S, <o_1, o_2, ..., o_n>) = Result(Result(S, <o_1>), <o_2, ..., o_n>). \quad (2)$$

To analyze our problem we introduced four operators classical in Block World (Nilson 1980):

- pickup(x) - block x is picked up from the table;
precondition list & delete list: ontable(x), clear(x), handempty
add list: holding(x)

- putdown(x) - block x is put down on the table;
precondition list & delete list: holding(x)
add list: ontable(x), clear(x), handempty

- stack(x,y) - block x is stacked on block y;
precondition list & delete list: holding(x), clear(y)
add list: handempty, on(x,y), clear(x)

- unstack(x,y) - block x is unstacked from block y;
precondition list & delete list: handempty, clear(x), on(x,y)
add list: holding(x), clear(y).

TRANSFORMATION TO LINEAR PROGRAMMING

Following (Bylander 1997) the transformation from planning to Linear Programming is based on mapping of conditions and operators in each plan step to variables. Truth values of conditions are mapped to 0 and 1 for the planning without incompleteness, and to any values between 0 and 1 for planning with incomplete information. The objective function reaches the maximum if the goal situation is true in the last step of planning.

As an example let us consider the problem 1 of planning in Block World environment with 4 blocks (called A, B, C, D). The goal is to decompose the initial state. It is assumed one STRIPS operator that moves the block x from the other block to the table, on(x) means that block x is on another block (or on the table), clear(x) means that there is no other block on block x :

move-to-table(x):
preconditions: on(x), clear(x)
del: on(x))
add: clear(y)

The goal is reached if the following conditions are true:

$$\text{clear(A), clear(B), clear(C), clear(D)} \quad (3)$$

Assume 2 steps of planning (states indexes are: 0, 1, 2). So (16+16+16) variables are needed for conditions for i = 0, 1, 2. In addition (12+12) variables are needed for operators (12 for transformation from state 0 to 1 and 12 from 1 to 2) for i = 0, 1 (see table 1).Then the objective function is:

$$\text{Max (clear(A)(2) + clear(B)(2) + clear(C)(2) + clear(D)(2)).} \quad (4)$$

If the goal is reached then the objective function is equal to 4 (4 conditions are true in the goal state). Constraints for Linear Programming problem are (Galuszka and Swierniak 2004):

- at most 1 operator can be applied in each planning step:

$$\Sigma \text{ move-to-table(x)(i)} = 1 \quad (5)$$

- operator can not be applied unless its preconditions are true:

$$\text{on(x,y)(i)} >= \text{move-to-table(x,y)(i)} \quad (6)$$

for all operators in each planning step,

$$\text{clear(A)(i)} >= \text{move-to-table(A,B)(i)} + \text{move-to-table(A,C)(i)} + \text{move-to-table(A,D)(i)} \quad (7)$$

$$\text{clear(B)(i)} >= \text{move-to-table(B,A)(i)} + \text{move-to-table(B,C)(i)} + \text{move-to-table(B,D)(i)} \quad (8)$$

$$\text{clear(C)(i)} >= \text{move-to-table(C,A)(i)} + \text{move-to-table(C,B)(i)} + \text{move-to-table(C,D)(i)} \quad (9)$$

$$\text{clear(D)(i)} >= \text{move-to-table(D,A)(i)} + \text{move-to-table(D,B)(i)} + \text{move-to-table(D,C)(i)} \quad (10)$$

for all planning steps.

Table 1. Variables of LP problem in each step

conditions	operators
on(A,B)(i)	move-to-table(A,B)(i)
on(A,C)(i)	move-to-table(A,C)(i)
on(A,D)(i)	move-to-table(A,D)(i)
on(B,A)(i)	move-to-table(B,A)(i)
on(B,C)(i)	move-to-table(B,C)(i)
on(B,D)(i)	move-to-table(B,D)(i)
on(C,A)(i)	move-to-table(C,A)(i)
on(C,B)(i)	move-to-table(C,B)(i)
on(C,D)(i)	move-to-table(C,D)(i)
on(D,A)(i)	move-to-table(D,A)(i)
on(D,B)(i)	move-to-table(D,B)(i)
on(D,C)(i)	move-to-table(D,C)(i)
clear(A)(i)	
clear(B)(i)	
clear(C)(i)	
clear(D)(i)	

Next group of constraints describes changes of the state after applying an operator. These are equality constraints:

$$\text{clear(A)(i+1)} = \text{clear(A)(i)} + \text{move-to-table(B,A)(i)} + \text{move-to-table(C,A)(i)} + \text{move-to-table(D,A)(i)} \quad (11)$$

and similar for blocks: B, C, D, and:

$$\text{on(A,B)(i+1)} = \text{on(A,B)(i)} - \text{move-to-table(A,B)(i)}. \quad (12)$$

Finally constraints for variables are needed: these should be mapped between 0 and 1 values. Transformation to Linear Programming results to the problem with 72 variables, 34 inequality constraints and 32 equality constraints. In each planning step there are 16 conditions and 12 possible operators.

In general case the size of LP problem is as follow (l denotes number of steps, n denotes number of blocks):
- the number of variables of LP problem:

$$(l + 1)(2n^2 + n), \quad (13)$$

- the number of inequality constraints:

$$l(n^2 + n + 1), \quad (14)$$

- the number of equality constraints:

$$l(n^2 + n), \quad (15)$$

so the transformation is polynomial.

If the LP problem is defined as:

$$\max_x \leftarrow f^T x \qquad \begin{aligned} Ax &\le b \\ A_{eq}x &= b_{eq} \\ 0 &\le x \le 1 \end{aligned} \quad (16)$$

then f defines the goal state of the planning problem, and the initial state is enclosed in Aeq and beq.

Example

To illustrate the transformation an example is shown. The problem (fig.1) is to decompose the initial state defined by the set: on(A,B), on(D,C), clear(A), clear(D). Matrices A and Aeq that correspond to the planning problem are sparse and their nonzero elements are illustrated in the fig.2.

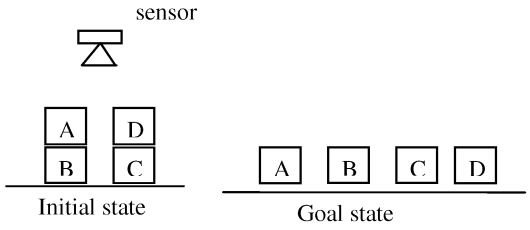


Fig.1. The initial and goal states for the example

The solution of Linear Programming algorithm is the set of variable values that gives optimal (maximal) value of objective function: variables from 69 to 72 are equal to 1. These corresponds to truth values of operators (variables 28 and 45): move-to-table(D,C)(0)=1 and move-to-table (A,B)(1)=1 that solve planning problem in 2 steps (Table 2).

Table 2. Solution of the Example.

conditions i = 0	actions i = 0	conditions i = 1	Actions i = 1	conditions i = 2
1.0000	-0.0000	1.0000	1.0000	0.0000
0.0000	-0.0000	0.0000	0.0000	-0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-0.0000	-0.0000	0.0000
-0.0000	0.0000	0.0000	-0.0000	-0.0000
0.0000	-0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	-0.0000	-0.0000	0.0000
-0.0000	0.0000	-0.0000	-0.0000	0.0000
1.0000	1.0000	0.0000	-0.0000	0.0000
1.0000		1.0000		1.0000
-0.0000		0.0000		1.0000
0.0000		1.0000		1.0000
1.0000		1.0000		1.0000

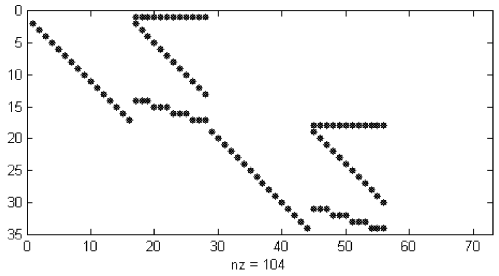
This result is, however, very optimistic: all variables (conditions) are 0's or 1's. In general case as a result we can receive any value from 0 to1. So the cost of the approach is

that algorithm can results in non-interpretable solutions for some initial states (what is followed by assumption P ≠ NP). The reason is because the discrete domain (truth or false) is transformed to continuous domain (LP program). For such cases the additional heuristic is proposed.

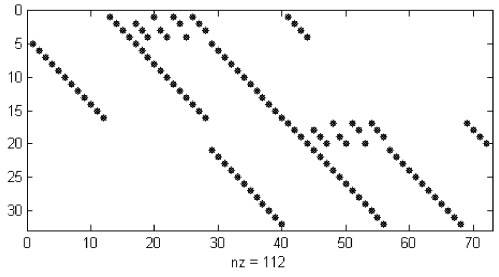
ADDITIONAL HEURISTIC

In the heuristic the variables values are treated as a 'truth degrees' and the algorithm scheme can be summarized as follow:

a) Solve the Linear Programming problem that corresponds to planning problem;
b) From the group of variables that correspond to operators in the 1st step choose the one with the biggest value of truth degree;
c) Assume the value of this variable equal to 1 (it means that this operator is choosed as the best proposition) and add this equality as an additional constraint to the LP problem;
d) Solve this more restricted problem;
e) Repeat steps b) to d) for the group of variables that corresponds to operators in the next step until result vector consists from only 0's and 1's.



A



A_{eq}

Fig.2. Matrices for linear program of the example

The number of repeats is limited by the number of steps. Fortunately, most of planning problems has the property that the number of steps is polynomial limited by the size of the problem (Baral et al. 2000). It follows that the linear transformation with additional heuristic remains polynomial.

An example of 'pathologic' planning problem is Towers of Hanoi (see e.g. A. Beck, M.N. Bleicher, D. W. Crowe, *Excursions into Mathematics*, A K Peters, 2000): in this case the number of operators growths exponentially with the problem size. This approach does not support such group of problems.

SIMULATIONS

To analyze computational efficiency of transformation of presented representations of STRIPS planning the 10 block decomposition problem has been implemented and solved using MATLAB® software. The assumed initial state is shown in Fig. 3. It is assumed 8 planning steps.

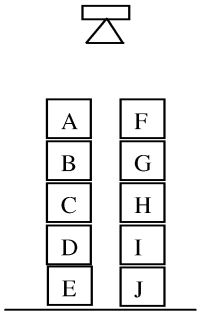


Fig. 3. The initial state for an exemplary problem.

In Fig.4. nonzero elements of matrices of 10 block 8 planning step block world problem are presented. The matrices of LP program are sparse (i.e. the number of non-zero elements is about 0,1% or less) and the large scale methods are very efficient in time (Zhang 1995).

The transformed problem consists of 1620 variables. Linear Programming problem has been solved using MATLAB's Large Scale Algorithm (method I), Large Scale Algorithm with additional heuristic (method II) and binary integer algorithm (method III – in this case variable values are limited to 0's and 1's)). For Large Scale Algorithm with additional heuristic the LP program was repeated 8 times with additional constraints (it correspond to the number of planning steps), because vector with many non-zero and non-one values was obtained as a result. In the fig.5 non-zero elements of solution vectors are shown: column 1 is for the first solution of LP program (many non-zero elements and most of them are also non-one), column 8 is for the last one (less non-zero elements and all of them are 1's).

The results of time efficiency are shown in the table 3.

Table 3. Efficiency in time

method:	I	II	III
time[s]:	2	4,07	594

CONCLUSION

Translation to Linear Programming is a heuristic that allows to reduce computational complexity of searching for the solution. That is because planning in the presence of incompleteness is usual at least NP-complete problem,

Linear Programming is polynomial-time complete problem and translation from STRIPS to Linear Programming is also polynomial (Bylander 1997). The cost of this approach is that algorithm can results in non-interpretable solutions for some initial states (what is followed by assumption P ≠ NP). This is because the solution of LP problem is the vector of possibly-continuous values from 0 to 1. So additional heuristic to receive the plan from the vector is needed. Binary integer algorithm (method III) returns the plan as a result, but the method is limited to only small sizes of planning problems. This is because the algorithm is exponentially efficient in time (this property is not a conclusion from the table but it is the property of the algorithm).

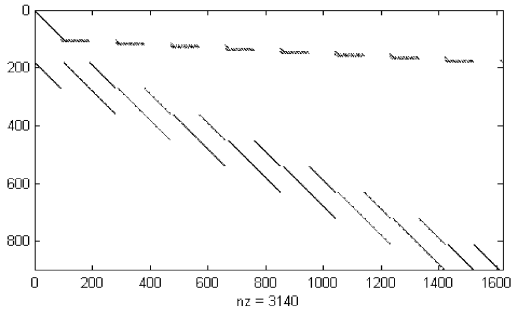


Fig.4. Nonzero elements of matrix Aeq of 10 block 8 planning step block world problem

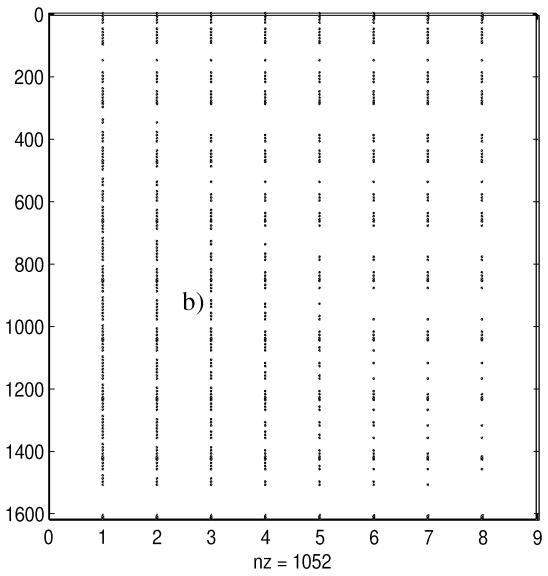


Fig. 5. non-zero elements of solution vectors.

Finally, results should be read as a comparison of efficiency of different introduced representations of Block World planning, taking into account that the transformation is polynomial and all methods (except III) are polynomial time complete. The conclusion that 4 seconds is needed to solve 10 block toy problem is wrong. So the heuristic is polynomial and that is why it reduces computational complexity of finding optimal solution of STRIPS planning problem, but only if this solution of transformed problem can be interpretable as a plan (the is the cost of transformation followed by the assumption P not equal NP).

Acknowledgement

This work has been supported by Silesian University of Technology BW/BK grant in the year 2007.

REFERENCES

Ambite, J.L, C.A. Knoblock. 2001. Planning by rewriting. Journal of Artificial Intelligence Research, 15: 207-261.

Backstrom, Ch. 1998. Computational Aspects of Reordering Plans. Journal of Artificial Intelligence Research, 9: 99-137.

Baral Ch., V. Kreinovich, R.Trejo. 2000. Computational complexity of planning and approximate planning in the presence of incompleteness. Artificial Intelligence, 122: 241-267.

Blythe J. 1999. An Overview of Planning Under Uncertainty. Pre-print from AI Magazine, 20(2), Summer 1999, pp. 37-54.

Boutilier C., Brafman R.I. 2001. Partial-Order Planning with Concurrent Interacting Actions. Journal of Artificial Intelligence Research, 14:105-136.

Bylander, T. 1994. The Computational Complexity of Propositional STRIPS Planning. Artificial Intelligence, 69:165-204.

Bylander, T. 1997. A Linear Programming Heuristic for Optimal Planning. In Proc. of AAAI Nat. Conf.

Chaczijan, L.G. 1979. A polynomial algorithm for linear programming. Dokł. Akad. Nauk SSSR, 244, 1093-1096.

Dougherty E.R., Ch.R. Giardina. 1988. Mathematical methods for artificial intelligence and autonomous systems, Prentice-Hall International, Inc. USA

Galuszka A., A. Swierniak. 2005. Non-cooperative game approach to multi-robot planning. Int. J. Appl. Math. Comput. Sci., Vol.15, No.3, 359-367.

Galuszka A. 2006. Block World Planning with Uncertainty and Sensing Actions as Integer Linear Programming Problem. [ed:] A. Cader, L. Rutkowski, R. Tadeusiewicz, J. Zurada: Artificial Intelligence and Soft Computing (ISBN 83-60434-09-3), IEEE Comp. Intell.Socielty - Poland Chapter, pp. 389-395.

Koehler, J.; J. Hoffmann. 2000. On Reasonable and Forced Goal Orderings and their Use in an Agenda-Driven Planning Algorithm. Journal of Artificial Intelligence Research, 12: 339–386.

Koehler, J.; K. Schuster. 2000. Elevator Control as a Planning Problem. AIPS-2000: 331-338.

Kraus, S.; K. Sycara; A. Evenchik. 1998. Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 104:1-69.

Nilson, N.J. 1980. Principles of Artificial Intelligence. Toga Publishing Company, Palo Alto, CA.

Slaney J., S. Thiebaux. 2001. Block World revisited. Artificial Intelligence 125: 119-153.

Smith, D.E.; D.S. Weld. 1998. Conformant Graphplan. Proc. 15th National Conf. on AI.

Weld, D.S. 1999. Recent Advantages in AI Planning. AI Magazine.

Weld, D.S., C.R. Anderson i D.E. Smith. 1998. Extending Graphplan to Handle Uncertainty & Sensing Actions. Proc. 15th National Conf. on AI, 897-904.

Zhang, Y.1995. Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, Technical Report TR96-01

CONTEX-BASED FUZZY CLUSTER-ORIENTED DECISION TREES

Zenon A. Sosnowski
Faculty of Computer Science
Bialystok Technical University
15-351 Bialystok, Wiejska 45A,
Poland
E-mail: zenon@wi.pb.edu.pl

KEYWORDS
Context-based fuzzy clustering, decision trees, expert systems.

ABSTRACT

In this study, we introduce a concept and design of decision trees based on information granules – multivariable entities characterized by low variability. As such granules are developed via context-based fuzzy clustering and play a pivotal role in the growth of the decision trees, we will be referring to them as CFC-decision trees. In contrast with "standard" decision trees in which we consider one variable (feature) at a time, this form of decision trees involves all variables that are considered at each node of the tree. The growth of the CFC-decision tree is realized by expanding a node of tree characterized by the highest variability of the information granule residing there.

INTRODUCTION

Decision trees (Quinlann 1986) are the commonly used architectures of machine learning and classification systems. They come with a comprehensive list of various training and pruning schemes, a diversity of discretization (quantization) algorithms, and a series of detailed learning refinements (Alexander and Grimshaw 1996; Cantu-Paz and Kamath 2000; Dobra and Gehrke 2002; Jain et al. 1999; Pedrycz and Sosnowski 2000; Pedrycz and Sosnowski 2005; Yildiz and Alpaydin 2001). In spite of such variety of the underlying development activities, we can easily witness several fundamental properties that cut quite evidently across the entire spectrum of the decision trees. First, the trees operate on discrete attributes that assume a finite number of values. Second, in the design procedure we choose one attribute at a time. Thirdly, decision trees in their generic version are predominantly applied to discrete class problems. The underlying conjecture is that data can be perceived as a collection of information granules (Pedrycz and Sosnowski 2000). Thus the tree becomes spanned over these granules treated now as fundamental building blocks. Furthermore fuzzy granulation helps link the discretization problem with the formation of the tree in a direct and intimate manner.

THE OVERALL ARCHITECTURE OF THE CLUSTER-BASED DECISION TREE

The architecture of the CFC-decision tree develops around fuzzy clusters that are treated as generic building blocks of the tree. For any context the training data set **X** is clustered into "c" clusters so that the data points that are similar are put together. These clusters are completely characterized by their prototypes. We start with them positioned at "c" top nodes of the tree structure. The way of building the clusters implies a specific way in which we allocate elements of **X** to each of them. In other words, each cluster comes with a subset of **X**, namely $\mathbf{X}_1$, $\mathbf{X}_2$, …,$\mathbf{X}_c$. The process of growing the tree is guided by a certain heterogeneity criterion that quantifies a diversity of the data (with respect to the output variable y) falling under the given cluster. Denote the values of the heterogeneity criterion by V_1, V_2, …, V_c, respectively (see also Figure 1).

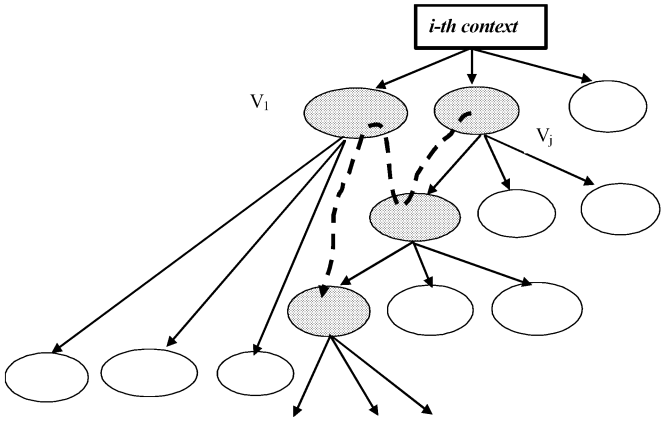


Figure 1: Growing a Decision Tree of the i-th Context by Expanding Nodes (that are viewed as clusters located at its nodes). Shadowed Nodes are those with Maximal Values of the Diversity Criterion and thus being Subject to the Split Operation.

Now we choose the node with the highest value of the criterion and treat it as a candidate for further refinement. Let i_0 be the one for which V_i assumes a maximal values, that is $i_0 = \arg\max_i V_i$. Again the resulting nodes (children) of node N_{io} come with their own sets of data. The process is repeated by selecting the most heterogeneous node out of all final nodes.

THE DEVELOPMENT OF THE TREE

In this section, we concentrate on the functional details and ensuing algorithmic aspects. The crux of the overall design is the clustering mechanism and the manipulations realized at the level of information granules formed in this manner.

Selecting information granules of contexts

It is the user who has to define what sort of information granules are of interest to him/her. We can anticipate two fundamental approaches to this problem: subjective definition of linguistic terms (fuzzy sets depending on needs of the potential users) and data driven definition of linguistic terms (Pedrycz and Sosnowski 2000). There, as in the previous case, we decide on the number of the linguistic terms using the same rationale as already underlined. The construction of the membership functions relies on the data. The essence is to distribute the terms (fuzzy sets) in such a way so that the σ-count (cardinality of fuzzy sets) computed over each of them is equal

$$\sigma(A) = \sum_{k=1}^{N} A(y_k)$$

where y_1, y_2, ... are the values of the context variable occurring in the data. In other words, the method relies very heavily on the histogram of the data (more precisely, a histogram of the context variable) and distributes the fuzzy sets so that they "cover" the same fraction of all data. The advantage is in the direct link between the linguistic granules and the experimental data. The disadvantage is in eventual lack of semantics of the constructed terms, especially in cases when they tend to be "crowded" in some regions of the context variable because of high intensity of the data points there. The other option of making the linguistic terms closely related to experimental data would be to cluster data and project the prototypes obtained in this way on the context variable. Again, this approach may exhibit the already mentioned weakness producing a lack of semantic integrity of the resulting information granules.
The details are contained in Figure 2. Given a collection of fuzzy sets (triangular and parabolic membership functions), we determine points of intersection of the two adjacent terms and form sets (intervals).

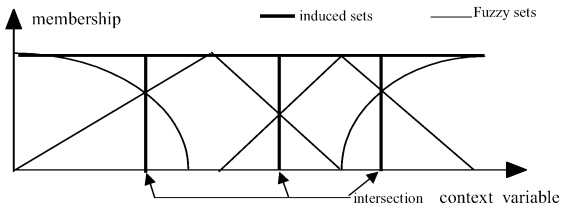


Figure 2: Constructing Linguistic Terms – Classes (thin line) and their Induced Interval-Valued Counterparts (solid line).

Context-based fuzzy clustering

Context-based fuzzy clustering is a core functional part of the overall tree. It builds the clusters and provides its full description. We refer the reader to (Pedrycz 1998). The overall algorithm is summarized as a sequence of steps

Given: The number of clusters (c). Select the distance function ||.||, termination criterion e (>0) and initialize partition matrix $U \in \mathcal{U}$. Select the value of the fuzzification parameter "m" (the default is m=2.0)
1. Calculate centers (prototypes) of the clusters

$$v_i = \frac{\sum_{k=1}^{N} u_{ik}^m x_k}{\sum_{k=1}^{N} u_{ik}^m}$$

i=1, 2, ..., c
2. Update partition matrix

$$u_{ik} = \frac{f_k}{\sum_{j=1}^{c} \left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{2/(m-1)}}$$

i=1, 2, ..., c, j=1, 2, ..., N

3. Compare U' to U, if termination criterion ||U' - U|| <e is satisfied then stop, else return to step (1) and proceed with computing by setting up U equal to U'

Result: partition matrix and prototypes

Node splitting criterion

The growth process of the tree is pursued by quantifying the diversity of data located at the individual nodes of the tree and splitting the nodes that exhibit the highest diversity. This intuitively appealing criterion takes into account the variability of the data, finds the node with the highest value of the criterion and splits it into "c" nodes that occur at the consecutive lower level of the tree, see Figure 3. The growth of the tree is controlled by conditions under which the clusters can be further expanded (split), cf. (Pedrycz and Sosnowski 2005).

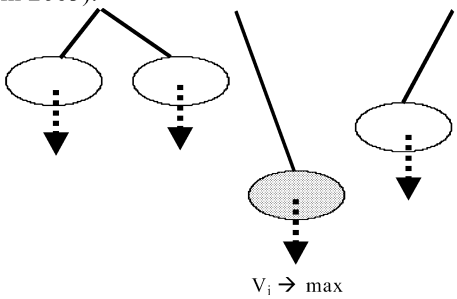


Figure 3: Node Splitting Controlled by the Variability Criterion V_i.

THE USE OF THE TREE IN THE PREDICTION MODE

Once the CFC-decision tree has been constructed, it can be used to classify a new input ($\mathbf{x}$) or predict a value of the associated output variable. In the calculations, we rely on the membership grades computed for each cluster as follows,

$$u_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \mathbf{x} - \mathbf{v}_i \|}{\| \mathbf{x} - \mathbf{v}_j \|} \right)^{2/(m-1)}}$$

where $\|\mathbf{x} - \mathbf{v}_i\|$ is a distance computed between $\mathbf{x}$ and $\mathbf{v}_i$. The calculations pertain the leaves of the CFC-tree so for several levels of depth we have to traverse the tree first to reach the specific leaves. This is done by computing $u_i(\mathbf{x})$ for each level of the tree, selecting the corresponding path and moving down, see Figure 4. At some level, we determine the path i_0 where $i_0 = \arg\max_i u_i(\mathbf{x})$. Once at the i_0-th node, we repeat the process that is determine j_0, $j_0 = \arg\max_j u_j(\mathbf{x})$ (here we are dealing with the clusters at the successive level of the tree).The process repeats for each level of the tree.

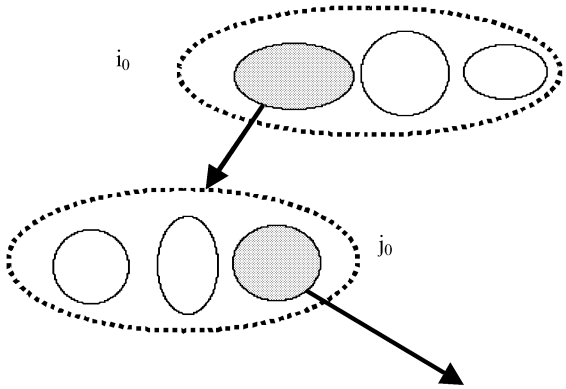


Figure 4: Traversing a CFC-decision tree: an Implicit Mode.

CONCLUSIONS

The CFC-decision trees are classification constructs that are built on a basis of information granules – fuzzy clusters. The way in which these trees are constructed deals with successive refinements of the clusters (granules) forming the nodes of the tree. When growing the tree we split the nodes (clusters) into granules of lower diversity (higher homogeneity). In contrast to C4.5-like trees, we use all features once at a time and such development approach promotes more compact trees and a versatile geometry of the partition of the feature space.

ACKNOWLEDGMENTS

Support from the Bialystok Technical University (grant W/WI/6/05) and 3 T11F 011 30 grant is gratefully acknowledged.

REFERENCES

Alexander W.P. and Grimshaw S. 1996. "Treed regression", *J. of Computational and Graphical Statistics*, 5, 156-175.

Cantu-Paz E. and Kamath C. 2000. "Using evolutionary algorithms to induce oblique decision trees", In: D. Whitley, D. E. Goldberg, E. Cantu-Paz, L. Spector, L. Partnee & H.-G. Beyer (eds.), *Proceedings of the Genetic and Evolutionary Computation Conference 2000*, pp. 1053-1060, San Francisco, CA: Morgan Kaufmann Publishers.

Dobra A. and Gehrke J. 2002. "SECRET: A scalable linear regression tree algorithm", *SIGKDD'02* Edmonton, Alberta, Canada.

Jain A.K. et al. 1999, "Data clustering: a review", *ACM Computing Surveys*, Vol. 31, No. 3, September 1999, 264 – 323.

Pedrycz W. 1998, "Conditional fuzzy c-means", *Pattern Recognit. Lett.*, vol. 9, pp.601-612.

Pedrycz W. and Sosnowski Z.A. 2000. „Designing decision trees with the use of fuzzy granulation", *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, vol. 30, 151-159.

Pedrycz W. and Sosnowski Z.A. 2005, "C-Fuzzy Decision Trees", *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Vol. 35, No 4, p. 498-511.

Quinlann J.R. 1986, "Induction of decision trees" , *Machine Learning* 1, 81-106.

Yildiz O.T. and Alpaydin E. 2001. "Omnivariate decision trees", *IEEE Transactions on Neural Networks*, vol. 12, no. 6, 1539-1546.

ZENON A. SOSNOWSKI was born in Bialystok, Poland. He received the MS degree in mathematics from University of Warsaw in 1976, and the PhD degree in computer science from the Warsaw University of Technology in 1986. He has been with the Technical University of Bialystok since 1976, where he is an Assistant Professor at the Faculty of Computer Science. In 1988/89 he had been at the Delft University of Technology in the Netherlands for 5 months. He spent two years (1990-1991) with the Knowledge Systems Laboratory of the National Research Council's Institute for Information Technology in Ottawa, Canada. His research interests include artificial intelligence, expert systems, approximate reasoning, fuzzy sets, and knowledge engineering. He is a member of the IEEE, and the IEEE Systems, Man, and Cybernetics.

NEURAL NETS

FORCE-BASED VISUALIZATION FOR COMPETITIVE LEARNING METHODS

Dusan Fedorčák
Ivo Vondrák
Department of Computer Science, Faculty of Electrical Engineering and Computer Science
VSB-Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
E-mail: dusan.fedorcak@vsb.cz, ivo.vondrak@vsb.cz

KEYWORDS
Neural network, Visualization, Physical model,
Classification, Self-organization

ABSTRACT

Competitive Learning methods such as Hard Competitive Learning or Neural Gas are commonly used for distributing certain number of reference vectors (neurons) in the input space. This distribution is utilized for vector quantization or feature mapping. Another common goal of those methods is clustering or classification. Well arranged visualization may be essential because in many cases human interpretation of resulting structures is needed. High-dimensional input space can be general issue when the suitable visualization is developed. Force-based graph drawing methods may be possible solution of this issue.

INTRODUCTION

The purpose of our research is to develop a tool for drawing structures generated by some competitive learning methods. In this article we will focus on using force-based (or force-directed) algorithm for visualization purposes. Since this algorithm is mostly designed for graph drawing, we will use it for competitive learning methods where a topology is generated. The Growing Neural Gas (Fritzke 1995) method is one of them and achieved results when the force-based layout is used, greatly improve interpretation of the structure generated by the Growing Neural Gas (GNG) method. In this article we will shortly describe the GNG method algorithm and then a suitable force-based algorithm will be presented. Finally, advantages and disadvantages of this approach will be discussed.

GROWING NEURAL GAS

The GNG method is derived from the Neural Gas (Martinetz and Schulten 1991) model and the main principle consists in mixing the Neural Gas approach with abilities of the growing structures. The main aim of all Neural Gas methods is describing the distribution of given dataset, thus these algorithms are frequently used for clustering or classification tasks. The GNG method extends Neural Gas algorithm by growing ability and it also generates a topology (Fritzke 1995) which can be utilized in visualization process. Simplified algorithm of the GNG method can be described as follows.

Simplified GNG algorithm

Start with small number of reference vectors (neurons). Take one pattern from the input set and find the closest neuron (winner). Make connection between the closest neuron and the second nearest one. Increase the local error attribute of the winner. Adapt the winner and its direct neighbors by moving them towards the input pattern. Manage edges and remove the old ones. Manage some other attributes such as stopping criterion, then choose another input pattern and start next iteration. After a fixed number of iteration grow the network in term of adding new neuron to position where the local error is the biggest.

Thus we are not presenting any modification nor the improvement of the GNG method in this article, the exact description of the algorithm is not needed here. Full description (Fritzke 1997) defines every step of the learning process and also contains possible improvements or modifications.

Two-Dimensional GNG Example

After several experiments we are able to discover some basic characteristic of the GNG method. During the learning phase generated network occupies given input space in a short time and then the method angles on more precise distribution by growing into a dense network. This behavior can be observed on simple example run on two-dimensional dataset (Figure 1).

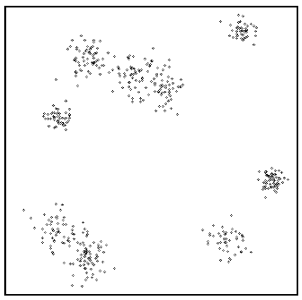


Figure 1: Two-Dimensional Dataset

Results are shown below (Figure 2). Generated network after 12000 iterations is shown on the left side and the same network after 40000 iterations is shown on the right side. Figures were generated using the GNG applet 1.5 (Loos and Fritzke 1997).

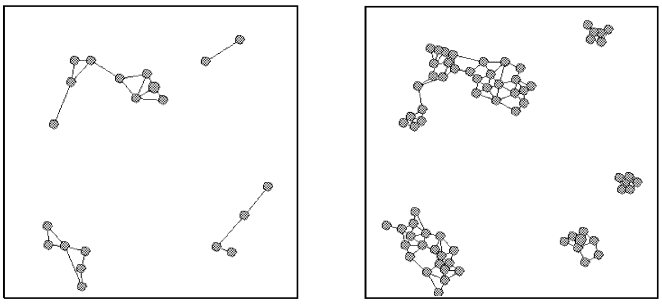


Figure 2: GNG Network 12000 Iterations (left) and 40000 Iterations (right)

Three-Dimensional GNG Example

Another example shows the GNG method in 3D space. The input dataset contains 2000 members randomly chosen from specified part of space (Figure 3). Dataset distribution can be imagined as the 'H' character twisted along the X axis.

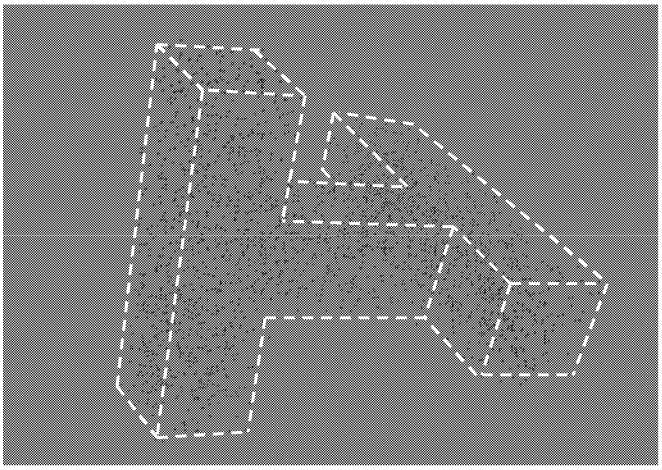


Figure 3: Three-Dimensional Dataset

The network generated by the GNG algorithm after 7000 iterations is shown below (Figure 4). As we mentioned before the method can describe dataset distribution within relatively shot time (iterations). The dataset shape is clearly visible.

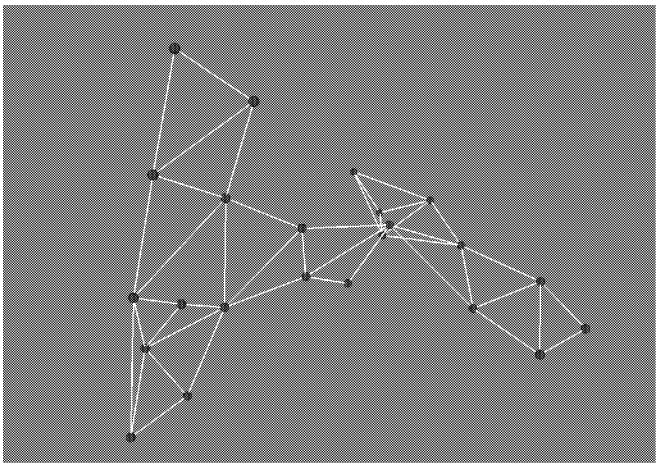


Figure 4: 3D Example of GNG (7000 iterations)

The same network after 60000 iterations is shown on next figure (Figure 5). Neurons (reference vectors) are uniformly distributed through whole dataset.

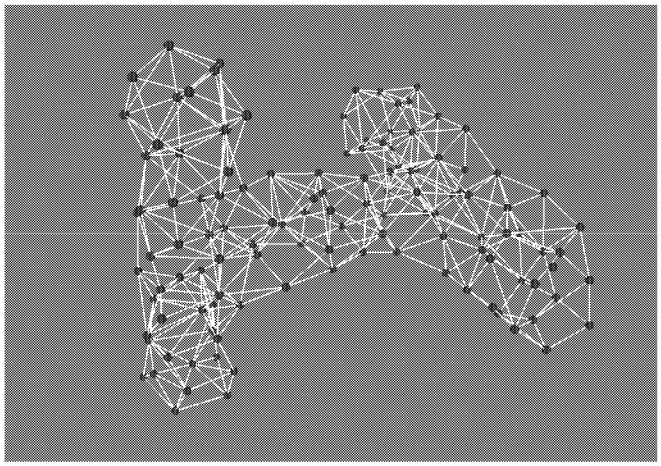


Figure 5: 3D Example of GNG (60000 iterations)

VISUALIZATION OF GROWING NEURAL GAS

As we shown above, presented test cases can be visualized in a quite simple way and this simple visualization can be widely used when the algorithm is configured for specific use or the method modification is tested. Evidently, datasets up to three dimensions will not make any problem when developing useful visualization. However, there is possibility of high-dimensional datasets when we move to practical use of the Competitive Learning Methods.

There is no need for visualization in some cases. If desired goal is vector quantization, the only utilized result is the winning neuron identification. Otherwise, if the goal is clustering the input dataset, or the preprocessing phase for the classification task, well arranged visualization may be essential for interpreting achieved results. There is also another reason for developing visualization methods for high-dimensional datasets. The classification problem on the low-dimensional data (e.g. up to 3D) can be resolved without any aid of artificial neural networks. When we move into the higher dimension, the visualization of pure dataset is not possible, so we have to find some way to lower the dimension of the result and utilize it in our visualization tool.

The main idea we followed when developing our visualization tool is that the classification task can be made by human recognition capabilities (which are still better then any artificial recognizer) when suitable aids are provided. A good visualization of the network generated by the GNG algorithm may be significant aid. Because there is no trivial way how the high-dimensional reference vectors can be visualized in 3D space (except some special cases like image recognition), we are using the reference vector (e.g. position in dataset space) only as a secondary value. The primary result we are interested in is the topology (e.g. the graph generated by the GNG method). If we assume this condition, the visualization problem is nearly reduced to the graph

plotting problem and the force-based algorithms can handle this situation with more then satisfying results.

FORCE-BASED ALGORITHMS

The Force-Based Algorithms (FBA) belong to the graph drawing algorithms class (Brandes 2001; Cruz and Tamassia 1999). The main idea of the FBA is constructing pseudo-physical model of a graph. Then the behavior of this model is simulated. The simulation generally converges to balanced state (e.g. state with minimal energy). When this state is reached, the graph is drawn. The physical model that we used is quite simple. Graph nodes are modeled as electrically charged particles and the edges between them are modeled as springs. Every particle is modeled as a mass point having adequate attributes (position, velocity, mass etc.). At every simulation step, resulting force is determined for every particle. Then the motion laws are applied. The simulation we used is based on simple Euler integration.

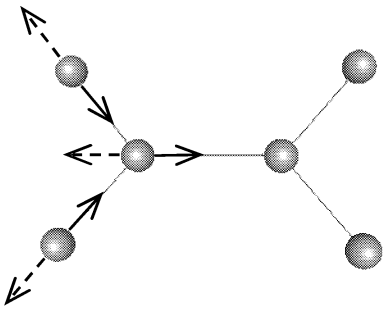


Figure 6: Simple Graph in Balanced State

Forces Definition

There are two basic forces applied on every particle. The repulsive force is similar to Coulomb's force and springs behavior e.g. the attractive force can be described by Hooke's law. In result, all particles are repulsing from each other and the connected ones are attracting their neighbors. Example of a simple graph drawing is shown on figure above (Figure 6). Attractive forces are shown as solid arrows; repulsive forces are shown as dashed arrows. Using more formal language, the forces described above can be defined as follows:

Resulting force affecting every particle:

$$\vec{F} = \vec{F}_r + \vec{F}_a \qquad (1)$$

Attractive force affecting particle x from its neighbors N based on Hooke's law, where m is number of neighbors:

$$\vec{F}_a = \sum_{i=0}^{m} k.d_i;\ i \in N_x;\ d_i = Pos_x - Pos_i \qquad (2)$$

Repulsive force affecting every particle x based on Coulomb's law, where n is number of particles in the graph:

$$\vec{F}_r = \sum_{i=0}^{n} c \frac{q_x . q_i}{\left\| \vec{d}_i \right\|^3} \vec{d}_i \qquad (3)$$

When the repulsive force (or the spring force) is evaluated, some constants have to be provided. These constants affect final graph appearance (for example c affects volume that the graph will acquire) and can be utilized to gather even more information from generated structure.

The graph generated by the GNG method has some undesired characteristics, so minor modification of the presented force-based algorithm is needed. As we mentioned before, the goal of the GNG method is clustering or classification problem, so we presume that generated graph will not be connected graph and it will contain number of isolated sub graphs. In result, the algorithm will produce a spatial-large drawing of the graph, where any sub graph will be repulsed from any other within a long distance (theoretically, within an infinite distance). Adding a force similar to the gravity may remove this issue. Figure below (Figure 7) shows the force functions for possible particle configurations.

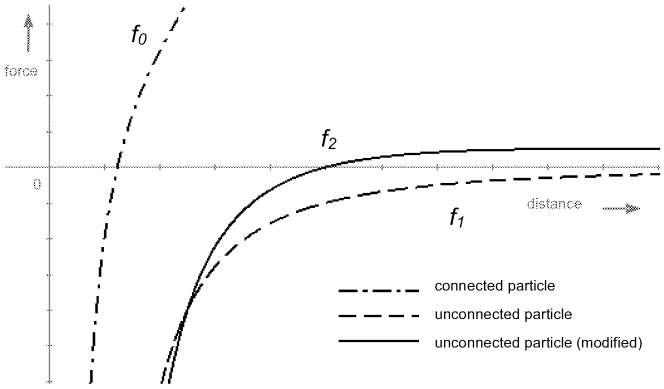


Figure 7: Force Functions

Dashed-dotted line represents resulting force function f_0 for a particle which is connected to another particle, the attractive force will prevail within a short distance. Dashed and solid lines represent the forces for an unconnected particle. While the force function f_1 contains only the repulsive force, force function f_2 combines the repulsive force and the attractive force. In result, when using modified repulsive force, the isolated sub graphs will not move further then defined maximum distance. The modified repulsive force can be defined as follows, where g is similar to gravitational constant and affects maximum distance between isolated sub graphs:

$$\vec{F}_r = \sum_{i=0}^{n} \left(c \frac{q_x . q_i}{\left\| \vec{d}_i \right\|^3} - \frac{g}{\left\| \vec{d}_i \right\|^2} \right) \vec{d}_i \qquad (4)$$

Example Force-Based Layout

The picture below (Figure 8) shows the GNG network processed on the same dataset as we used for 3D example (Figure 3). Force-based algorithm was used for graph layout. As we mentioned before no reference vector value has been used while generating this layout. The topology itself contains needed information and resulting layout corresponds with dataset distribution.

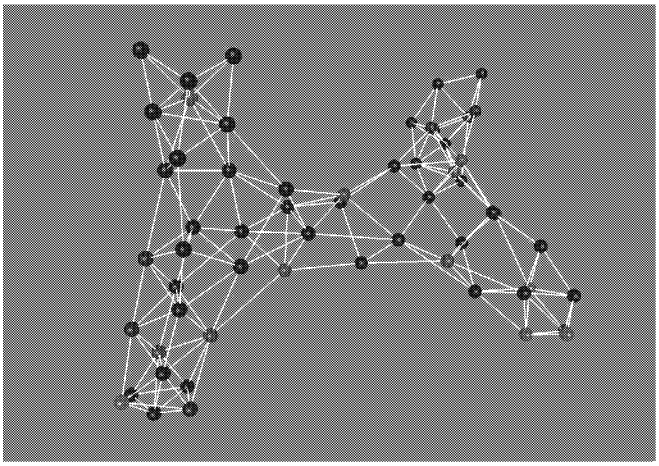


Figure 8: Force-Based Graph Layout

There is another noticeable advantage when we are mixing the force-based layout with the GNG learning process. The major disadvantage of the FBA is that the resulting graph may relax in a local minimum which is considerably worse then a global minimum (Brandes 2001). When the GNG learning process runs concurrently with the FBA relaxing process, this issue may be reduced due the network grows from small number of connected nodes into a dense graph. In this way, there is high probability of hitting the global minimum when the network is in the beginning of the learning process. Moreover, every new neuron generates a stress state in the relaxing process and the network is able to pull out from undesired local minima.

TEST CASES

As we presented above, the Growing Neural Gas method along with force-based layout can be used as preprocess for classification problem. Following text presents test cases where the dataset dimension is greater then three so there is no trivial way how to visualize reference vectors in dataset space.

Iris Plant Test Case

The first test case is well known Iris Plants Database (Newman et al. 1998). It contains 150 instances, each with four numeric attributes. The resulting network captured from our tool is shown below (Figure 9). At this point we are able to analyze the dataset distribution and try to identify possible classes. There is definitely one class represented by isolated sub graph. The topology of the other sub graph indicates that the input dataset may contain two or three different groups of patterns (robust branches connected with a thin bridge).

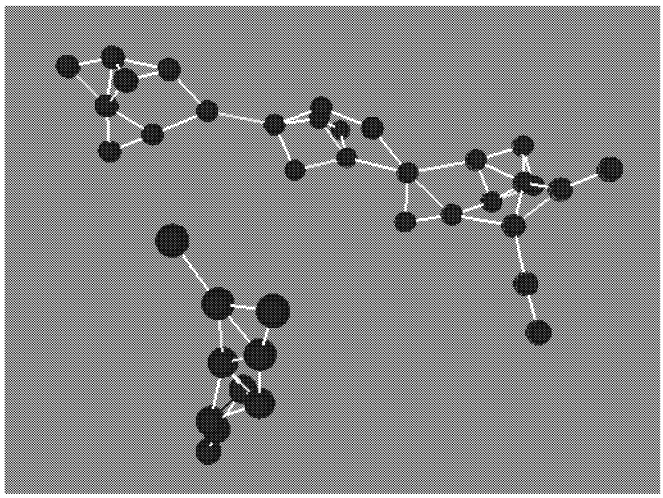


Figure 9: Iris Plant Test Case

Now, we can compare our prediction with known classification facts supplied from original dataset. The instances can be classified into the three different classes, according to the iris plant type. One class is linearly separable form the other two; latter are not linearly separable from each other. If we highlight each reference vector according to instances mapped on it during the learning process, we can validate our prediction (Figure 10).

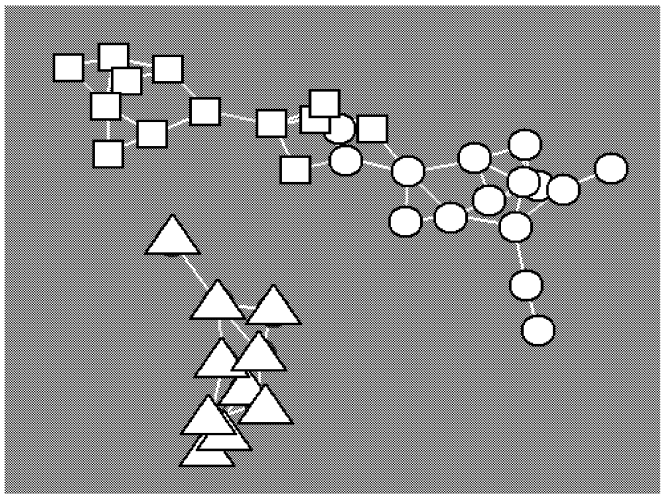


Figure 10: Iris Plant Test Case (classified)

A sub graph with triangle-highlighted point represents linearly separable class and the other sub graph represents latter classes.

Surface Type Test Case

Another test case is a dataset of images describing different types of surfaces like cement, brick wall, foliage etc. (Newman et al. 1998). Every pattern has 20 numerical attributes in terms of their image characteristics. The resulting graph drawing (Figure 11) shows again one probably linearly separable class, other classes can be identified as branches emanating from the graph core. Every class is presented as different shape.

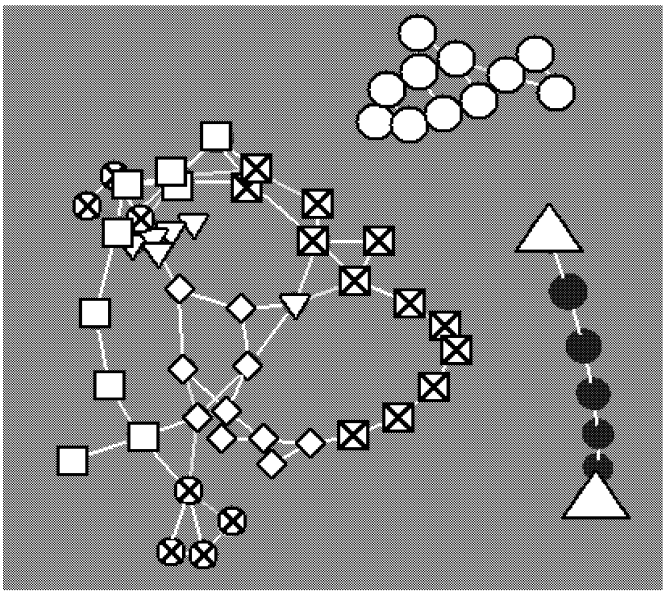


Figure 11: Surface Types Test Case

Noticeable situation arisen in this test case showing possible error in the input dataset. The GNG method generated string of neurons, where the inner nodes are not "winning" neurons in any case; they are only describing large empty hole in the input space. This means that some very different and possible error pattern was mapped to the outer neuron/s (triangles).

Hand-Written Digit Test Case

The last test case is also well-known MNIST hand-written digit database (LeCun et al. 1998a). This database contains 60000 instances of grayscale images. Every image is 28x28 pixels, thus pattern dimension is 28 squared. The result after approximately 40000 iterations is shown on last picture (Figure 12). The resulting graph drawing is not as well-arranged as in the cases before, but still there are recognizable parts of the graph which contains reference vectors corresponding with different types of hand-written digits. There is one advantage in the visualization of this test case; every reference vector can be visualized as an image, so we are able to prove GNG distribution in the input space.

CONCLUSIONS

In this text, we presented our approach for visualization of possibly high-dimensional structures generated by Growing Neural Gas method. We successfully used the force-based algorithm for this purpose and we presented some test cases showing generated structure drawings. We are also able to remove some issues of force-based layout such as locking in local minima but one major issue still remains. Force-based approaches consume a huge amount of time. Further studies have to be concentrated to removing or reducing this issue. For example it is possible to lower the priority of graph relaxing process radically when the graph is near balanced state. Another improvement may be found in utilizing parameters included in the FBA relaxing process, such as defining individual repulsive force between two graph nodes which will correspond with the reference vector position in the input space. In such manner we are able to gather even more information about structures generated with Competitive Learning methods like Growing Neural Gas.

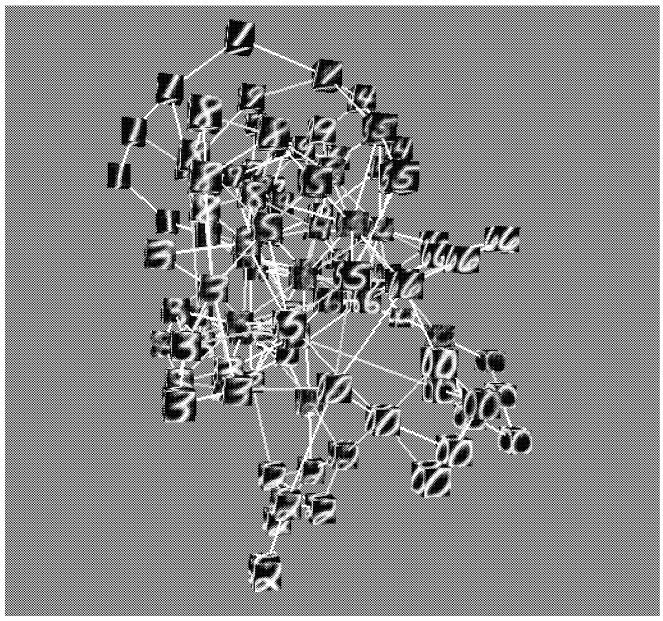


Figure 12: MNIST Hand-Written Digit Database Test Case

ACKNOWLEDGEMENT

This research has been supported by the Moravian-Silesian Region, Regional Authority of Czech Republic, project No. 01562/2006/RER.

REFERENCES

Brandes Ulrik "Drawing on Physical Analogies" M. Kaufmann and D. Wagner (Eds.): *Drawing Graphs: Methods and Models*, LNCS 2025, pp. 71-86, 2001. Springer-Verlag Berlin Heidelberg 2001

Cruz Isabel F., Tamassia Roberto "Graph Drawing Tutorial" pages 66-72 http://www.cs.brown.edu/~rt/papers/gd-tutorial/gd-constraints.pdf

Fritzke B. "A growing neural gas network learns topologies" in G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625-632. MIT Press, Cambridge MA, 1995.

Fritzke B. "Some Competitive Learning Methods" http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/

Loos Hartmut S., Fritzke Bernd "DemoGNG (Version 1.5)" http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG_2.html

Martinetz T. M., Schulten K. J. "A neural-gas network learns topologies" in T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 397-402. North-Holland, Amsterdam, 1991.

Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998) "UCI Repository of machine learning databases" [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Yann LeCun, Courant Institute, NYU, Corinna Cortes, Google Labs, New York. "THE MNIST DATABASE of handwritten digits" http://yann.lecun.com/exdb/mnis

EVALUATION OF FOREST TREE DISTRIBUTION MODEL USING ARTIFICIAL NEURAL NETWORKS

Salvis Dagis
Faculty of Information Technologies,
Latvia University of Agriculture,
Liela Street 2, Jelgava, Latvia
E-mail: salvis.dagis@llu.lv

KEYWORDS

Mathematical modeling, Artificial Neural Networks, Normal and Skew-Normal Distribution.

ABSTRACT

Forests cover up to 45% of the territory of Latvia and forestry is the most significant export sector in Latvia. In total forestry provides up to 14% of Gross Domestic Product. To make more precise measurements of their forest economical indicators, forest owners need a quality primary data processing. This allows tree distribution by diameter class using average forest valuation indicators. So far, one used to apply virtual datasheet mathematical algorithm by R.Ozolins that is based on assumption that tree distribution by diameter corresponds to that of normal. The research showed that this distribution features certain dissymmetry that depends on the tree diameter. As a result, a new algorithm was produced on the basis of artificial neural network, which proved to be twice more precise and effective.

INTRODUCTION

With the development of technology and computer capacity, the use of ANN (artificial neural networks) has become more actual and widely applied in such areas as medicine, finance, picture and sound recognition. ANN is successfully used also in data classification, clustering, optimisation and function approximation, which is one of the present research objects. Forest management planning is one of the most prospective branches where ANN can be used.

Regulations issued by the Cabinet of Ministers of the Republic of Latvia suggest regular auditing – at least once in 10 years each property is to submit its forest average valuation indicators according to tree species, such as average tree height (H), average tree diameter (D), cross-area per hectare (G). These data are basically applied in forest development analysis and forest statistics. To apply them to more precise processing of economical indicators, it is necessary to know the assortment distribution if the trees were cut off and sorted. From valuation indicators one knows the total wood yield per hectare (V), but it doesn't give any idea of the economic value of the forest. Timber market price is determined by the assortment, which normally is divided into 5 groups according to the beam spigot diameter.

It is possible to calculate the assortment for the area if one knows the height and diameter of every tree of each species or the tree distribution by the diameter class (Ozolins 2002).

Measuring every tree (precise valuation) is performed only before the commercial order for the main cut, since the government demands to submit the information about bulk forest usage. In other cases, they do not measure every single tree as it is a resource-consuming and expensive process. At the same time, forest owners go into forest purchasing, selling and management. They need to know the current forest value as well as its future potential. One of the problems for forest owners is the evaluation of the latter. The owners would like to know the forest value without even entering the forest, therefore, the value has to be told from the valuation data. For this purpose, a virtual datasheet algorithm (MATH_OZO) has been elaborated recently (Ozolins 2004) that is based on the assumption that tree distribution by the diameter class corresponds to that of normal. The forest owners who use this algorithm for their forest evaluation often inquire how reliable it is and how precise the calculated data are. To answer these questions and possibly improve the precision of the algorithm it is necessary to compare the calculated datasheet data to those taken in the forest proper (precise valuation data).

This research is about the accuracy evaluation of the existing virtual datasheet algorithm, as well as about the elaboration of a new method of tree distribution by the diameter class. The method is based on using ANN. As the basis for evaluating the existing algorithm and elaborating the new one, the author used precise valuation data. The valuation has been held from 2000 to 2007 in different regions of Latvia. Total volume of measuring is 3695 sampling plots with 882 750 trees measured. Each tree was measured using one of the following electronic measuring instruments: Masser 2000GR or Masser Excalliper. With the help of these instruments it was possible to determine the tree diameter and height, while the tree species was put in manually. From the measured data, the wood yield, cross-area and the number of trees per hectare (N) were calculated.

The purpose of the research is to elaborate the upgrade for MATH_OZO algorithm with ANN so that the former is faster and more accurate.

MATERIALS AND METHODS

To determine assortment distribution that is used to calculate the economic indicators the datasheet is exploited. It consists of tree count record according to their diameter. Traditional datasheet diameter distribution step is 4 cm. it is exactly the same in MATH_OZO virtual datasheet, or the datasheet calculated from average valuation data.

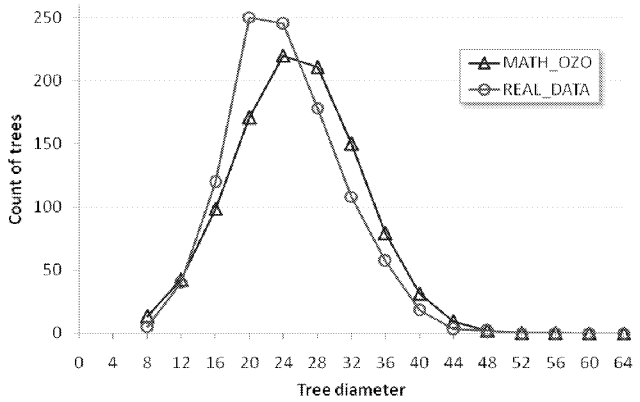


Figure 1: MATH_OZO result comparison to real data

Tree distribution by the diameter class can be seen in Figure 1, where one can also notice that real data (REAL_DATA) distribution is similar to the normal (1). In this case, the normal distribution is interpreted also by MATH_OZO algorithm. There is some difference between real and calculated tree distribution, which appeared as the result of determination of normal distribution dissymmetry and dispersion (S). At the moment, the dissymmetry indicator in the used MATH_OZO algorithm is not taken into account.

VIRTUAL DATASHEET GENERATING ALGORITHM

For several years already, the virtual datasheet generating algorithm by R.Ozolins (Ozolins 2004) is used in Latvia for forest evaluation. The algorithm was elaborated using 38 precise valuation datasheets. The algorithm is based on assumption that tree distribution according to their diameter corresponds to that of normal (1).

$$f(x,\mu,s)=\frac{1}{s\sqrt{2\pi}}e^{\frac{-(x-\mu)^2}{2s^2}} \quad (1)$$

To determine the normal distribution, one uses modal diameter (2), not average diameter.

$$D_{mod}= 1.22027*D-8.46487 \quad (2)$$

Modal diameter is the one with the greatest frequency of the distribution by diameter classes. Modal diameter shows the first restriction for using the algorithm – it cannot be applied if the average diameter is less than 7 cm, for then the acquired value, i.e. diameter becomes negative. Therefore, also virtual datasheet diameter starts from 8 cm (Figure 1). Modal diameter with lesser D and bigger S assets produce the deviation of the normal distribution towards the negative assets.

$$S=0.06832*D+5.69188 \quad (3)$$

As another restriction, one should mention S value limits, which according to this algorithm (3) – even if all three recommended perturbations are applied (Ozolins 2004) – cannot be less than 3.5 and bigger than 15, however its range as it is seen from the real data is within the limits from 9 to 12. Perturbations are iterations where primarily denoted values for S and tree contour lines are altered to positive and negative directions with coefficient 0.25 to find the correspondence between the contour line and distribution by diameter that would result in equality of the acquired and real forest yield. Moreover, this perturbation approach that varies 2 parameters (tree contour line and asset S) to get the total of accurate yield figure yet is not the ideal solution as it provides an infinite number of correct final variants.

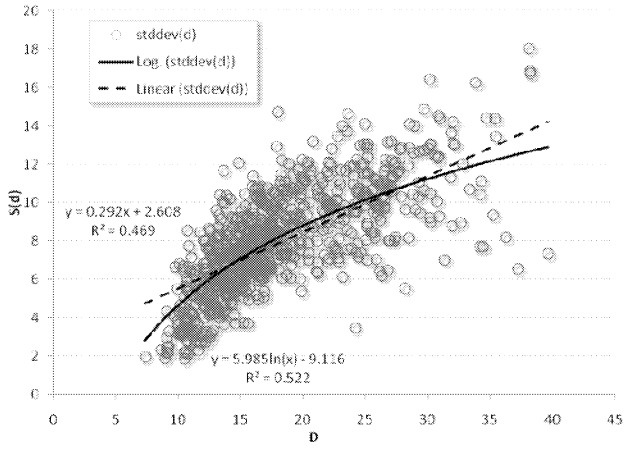


Figure 2: Diameter dispersion

In Figure 2, one can see that the normal distribution parameter S does not depend on D, for the dispersion is too great. S(D) can be described neither with linear, as it is done in case with MATH_OZO (3), nor with logarithmic equation, as the coefficient of correlation R^2 is too small.

DIAMETER DETERMINATION USING ANN

The existing mathematical MATH_OZO models for tree distribution according to the diameter are rather complicated and their modification is problematic, though their working speed could be better (it is problematic with large data amounts), yet the elaboration of the new ones calls for thorough mathematical knowledge. Taking into consideration the above, the author decided to find an alternative solution. It is going to be based on ANN, which provides the possibility of getting equal results in a simpler way. Neural networks are models that are designed to imitate the human brain through the use of mathematical models.

Using ANN has the following advantages:

- Compact knowledge reflection in the form of weight and activation function matrix;
- Very fast and simple functioning due to the previously mentioned feature;
- They can perform with noisy or missing data and still give good results with yet unseen data;
- They are trained inductively by the training data set;

- They are capable to process nonlinear functionality, which is especially essential working with real nature data.

However, ANN features some drawbacks as well:

- Its training session can sometimes be rather prolonged and may not converge the desired result;
- Using accidental assets for weight initialisation, there is a chance of coming to various problem solutions;
- ANN topology is created in the empirical way, so sometimes in order to come to the desired result it is necessary to take several attempts;
- They have a limited result interpretation, which occasionally prevents one from using them;
- It is complicated to supply ANN with new knowledge and equally difficult to pass the existing knowledge on to another ANN.

The suggested solution features some provisions to make it faster, more precise and simpler for everyday use. There are 3 solution alternatives that can be accomplished with ANN.

The first alternative is how to choose the distribution frequency across diameter classes with ANN target values. To secure the diameter value range 8-64 cm distribution, a 4 cm optimal diameter distribution step has been determined. This distribution requires from ANN 15 output neurons. To fulfil this aim it is necessary to have a very large data amount, as forest valuation seldom features marginal values, especially those of diameters. Among the available data, some species lack measurements for thick trees distribution classes. Therefore, in the ANN training process corresponding outputs will always be given values equal to zero, which will produce errs in the result at times when these data is wholly represented. Training ANN using available data, there were noticed data imperfections that made a negative impact on the desired result, for ANN successfully adjusted itself to data errors. So, this kind of approach was considered to be applicable only in cases when there is sufficient and quality data available, but not with the data like those used in the present research.

Another alternative is to use the function of normal distribution like it is done in case with MATH_OZO, and use ANN for dispersion parameter S determination. Analysing precise valuation data, it was noted that distribution across diameter classes generally corresponds to the normal distribution, but there are instances when dissymmetry is noticed.

The third alternative is to determine the dissymmetry coefficient for the normal distribution and use it to complete the normal distribution function, establishing the dispersion parameter S with ANN. In this case, one can use the modified normal distribution function (4), which is called 'skew-normal distributions function' (Azzalini and Capitanio 1999). Normal distribution asymmetric function is acquired as double multiplication of the function of standardised normal distribution (5) and integrated standardised normal distribution cumulative function (6). It should definitely be added that the normal distribution cumulative function is modified, its integral upper marginal value multiplied with asymmetry coefficient value.

$$f(S,\alpha) = 2\varphi(S)\cdot\phi(S,\alpha) \quad (4)$$

$$\varphi(S) = e^{\left(\frac{-x^2}{2}\right)} \Big/ \sqrt{2\pi} \quad (5)$$

$$\phi(S,\alpha) = \int_{-\infty}^{S\cdot\alpha} \varphi(t)\,dt \quad (6)$$

To evaluate the acquired ANN results using real precise valuation data, the former were transformed into distribution according to diameter levels. Within each level the quadratic error was calculated, then the latter was calculated from all levels within one sample data set. To get an objective error evaluation, the average error was calculated also among all sample data sets. The average error later was compared within various calculation solutions.

RESULTS

Using ANN, the following results were acquired:

- The speed and accuracy increased;
- The application area of the newly-elaborated algorithm is much wider than that of the existing MATH_OZO algorithm:
  - It is possible to change the datasheet diameter distribution step;
  - Perform re-training using specific data if necessary;
  - Train with other species;
- The algorithm got probated on the 4 mostly common Latvian tree species (the mentioned species comprise 89% of the total forest area) and different training sample amounts. In the result, a training error was acquired (see Table 1);

Table 1: ANN training data and results

	Number of training samples	Training errors (%)
Pine	379	2.68
Spruce	621	2.47
Birch	578	2.02
Black alder	338	3.51

ANN was created in C language on the basis of *Fast Artificial Neural Network Library* (http://leenissen.dk/fann/). The full ANN training code can be seen below (Figure3).

```
#include "fann.h"
int main(int argc, char **argv)
{
    const unsigned int num_input = 5;
    const unsigned int num_output = 1;
    const unsigned int num_layers = 3;
    const unsigned int num_neurons_hidden = 10;
    const float desired_error = (const float) 0.0001;
    const unsigned int max_epochs = 2000000;
    const unsigned int epochs_between_reports = 10000;
    if (argc==3)
    {
        struct fann *ann = fann_create_standard(num_layers,
                           num_input, num_neurons_hidden, num_output);
        fann_set_training_algorithm(ann,FANN_TRAIN_INCREMENTAL);
        fann_set_activation_function_hidden(ann, FANN_ELLIOT);
        fann_set_activation_function_output(ann, FANN_ELLIOT);
        fann_train_on_file(ann, argv[1], max_epochs,
                epochs_between_reports, desired_error);
        fann_save(ann, argv[2]);
        fann_destroy(ann);
    }
    else
    {
        printf("Usage:\n\t%s inp_train.data out_train.net\n",argv[0]);
    }
    return 0;
}
```

Figure 3: ANN training code in C language

It is worth emphasising that the amount of neurons in the hidden layer was defined in experimental way, as the literary sources analysis didn't provide any methodology for determining optimal neuron amount. The most appropriate activating function for this ANN is the function by David Elliott (3), where X is input value, Y – output value, and S – steepness.

$$y = ((x*s) / 2) / (1 + |x*s|) + 0.5 \quad (7)$$

This activating function was more stable: it converged faster and there was no divergence observed, like it used to be with the traditional *sigmoid* function (4)

$$y = 1/(1 + \exp(-2*s*x)) \quad (8)$$

Creating ANN structure, the author chose a simpler network topology - feedforward neural network) (Figure 4)

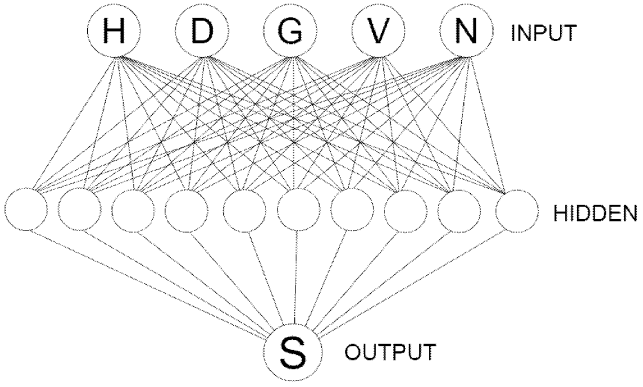


Figure 4: Created ANN structure

The number of neural network layers was determined in experimental way, where single-layer network (INPUT-OUTPUT) showed very poor results in the training process, while two-layer network (INPUT-HIDDEN-OUTPUT) performed quite satisfactory. There was an attempt made to add another HIDDEN layer, which gave no essential improvements in training accuracy. Thus, the three-layer network was approved as suitable for the case. This network can be depicted using the following mathematical structure (9).

$$S = f_o\left(\sum_{h=1}^{H} W_h \cdot f_h\left(\sum_{i=1}^{I} W_i \cdot X_i + b_h\right)\right) + b_o \quad (9)$$

X_i – input layer value;
i – input layer neuron index (i=1..I);
h – hidden layer neuron index (h=1..H);
f_0 – activating function (7)in the output layer;
f_h – activating function (7) in the hidden layer;
W_h – weight values from input to hidden layer;
W_o – weight values from the hidden layer to output layer;
b_h, b_o, – constant weight values (bias) for hidden and output layers;

During the ANN training process the input parameters were altered, beginning with just primary valuation data (H,D) and finishing with adding the calculated data (G,V,N), as the result of which high level of precision and faster network training were acquired (Figure 5).

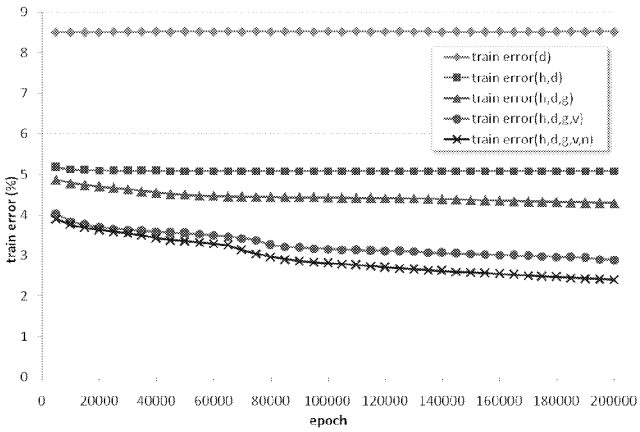


Figure 5: ANN training error evaluation

Carrying out the experiment with the option of input parameters, it was concluded that each added ANN parameter diminishes the training error. The error is faster converged, which means that each parameter is essential and contributes to higher network accuracy. As it is seen in Figure 6, the trained ANN weight intensity (implication) is very big among entire neurons, which is indicated by letters A-Z, where Z indicates the biggest weight value. The fifth input neuron (N) makes the least impact on ANN training accuracy.

```
Layer / Neuron 0123456789012345 6
L   1 / N    6 LzqBzH...........
L   1 / N    7 ZzZzHd...........
L   1 / N    8 xJzzKF...........
L   1 / N    9 KxtIce...........
L   1 / N   10 HkzZzB...........
L   1 / N   11 ZzZzbe...........
L   1 / N   12 zZCzzH...........
L   1 / N   13 ZzzEvb...........
L   1 / N   14 zZzZRI...........
L   1 / N   15 GzUzQG...........
L   1 / N   16 ...............
L   2 / N   17 ......NrTZRjzKIpk
L   2 / N   18 ...............
```

Figure 6: FANN network weight analysis

Taking into account the fact that the attempt to train ANN so that the dissymmetry coefficient is acquired failed, it was stated in experimental way by observing the dissymmetry dependence on average diameter, that the connections of all 4 observed species are similar (Figure 7). These were approximated using logarithmic equation. For the entire observed tree species the curve performs equally, i.e. dissymmetry is positive with younger thinner trees, while it turns into negative with trees with larger diameters.

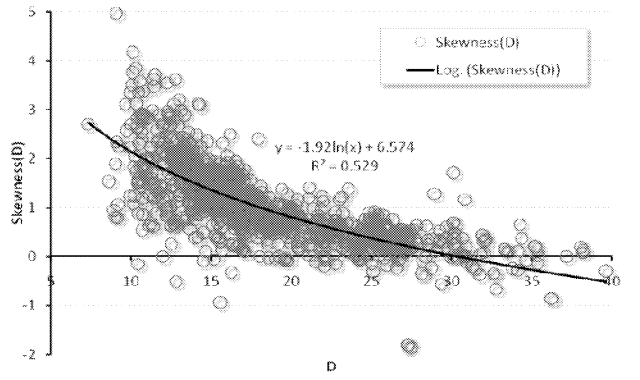


Figure 7: Dissymmetry coefficient dependence on the diameter (birch)

For the needs of algorithm testing every tenth precise valuation data set was taken, providing it wasn't used in the training process. The algorithm performance speed in calculating the diameter distribution classes was checked too during the testing period. The algorithms based on ANN gave results 92 times faster than the MATH_OZO algorithm. Such a speed increase can be justified by simple mathematical calculations (9), for MATH_OZO is a complicated and iterative algorithm.

Table 2: Comparison of acquired results with precise valuation data.

	Mean Square Error (MSE) comparison		
Species	MSE_MATH_OZO / MSE_ANN_S	MSE_MATH_OZO / MSE_ANN_S_SKEW	MSE_ANN_S / MSE_ANN_S_SKEW
Pine	1.36	1.67	1.23
Spruce	1.67	2.07	1.24
Birch	2.16	2.72	1.26
Black alder	1.79	2.00	1.12
AVERAGE	*1.75*	*2.12*	*1.21*

The acquired results can be seen in Table 2 which features the relations of average quadratic error among three methods:

- MSE_MATH_OZO – average quadratic error for MATH_OZO algorithm;
- MSE_ANN_S – average quadratic error using normal distribution function (1) with ANN-calculated parameter S;
- MSE_ANN_S_SKEW – average quadratic error using skew-normal distribution function (4) with ANN-calculated parameter S and asymmetry coefficient acquired using logarithmic equation (Figure 7);

The quadratic error for each method is calculated between real precise valuation data and the data acquired for each very method. As it is seen in Table 2, the ANN-calculated normal distribution S parameter gives on average 1.75 times more accurate results if compared to the existing MATH_OZO algorithm. In its turn, using skew-normal asymmetry function (4), the accuracy increases by 1.21 times in compared to normal distribution function without dissymmetry. Upon the whole, the best results are observed among MATH_OZO algorithm and skew-normal asymmetry function (4), where the accuracy is improved by an average 2.12 times, and with some species even up to 2.72 times.

All in all, ANN takes over OZO in terms of accuracy more than 2 times. Using the former for calculating various economical indicators will result in higher data reliability and the acquired data will be much close to the actual situation in the forest.

CONCLUSIONS

Using ANN, the accuracy increased by 2 times, calculation speed – by 92 times. The way is found to train ANN using other tree species.

Using post-training table of ANN neural structure analysis (weight intensity), it is possible to determine each input parameter's significance for the entire system.

The calculation accuracy during the ANN training process changes according to the amount of data.

For ANN training, it is necessary to have sufficient amount of data or that containing characteristics of all the general data sets.

Distribution by the diameter class depends on tree species, since using ANN trained on one species for another, a greater error in diameter distribution was encountered, if compared to real data.

Distribution by the diameter class can be more accurate if the ANN training takes into consideration such an aspect as tree growing conditions (they were not considered in the present research for the lack of data).

Distribution by the diameter class is characterised by dissymmetry that depends on the average diameter value.

The practical benefit from using ANN is the flexibility of this algorithm, its resistance to noisy and insufficient data, chance for complicated problem solution, as well as the possibility to solve nonlinear functionality.

REFERENCES

1. Azzalini A. and Capitanio A. 1999. Statistical applications of the multivariate skew-normal distribution. J. Roy. Statist. Soc., B 61, 579-602.
2. Dagis S., Arhipovs S., "The static model of Latvian forest management planning and capital value estimating", Proceedings, "8th International Conference on Enterprise Information Systems", Cyprus, Paphos, 2006.
3. Dagis S., Arhipovs S., Dubrovskis D., - The growth of trees motion mathematical models and their adaptation the Latvia circumstances, Lithuania, Kauno, 2006
4. Dagis S., „Information system requirement analysis and specification in forest management planning process", 9th International Conference on Enterprise Information Systems, Funchal, Madeira – Portugal, 2007
5. Ozolins R. 2004. Tree trunk distribution by the diameter class. Proceedings of the 5 international Conference – Latvian academy of pedagogy.
6. Ozolins R. Forest stand assortment structure analysis using mathematical modelling. Forest structure and growth – Tartu, 2002. – (Forestry studies XXXVII) p. 33-42.

AGENT BASED SIMULATION

AGENT BASED PLANNING

AGENT PLANNING AND SCHEDULING IN DISTRIBUTED MANIFACTURING SYSTEMS

Kamel Benaissa
Daniel Diep
LGI2P-EMA, Parc Scientifique G. Besse,
30035 Nîmes Cedex 1, France
E-mail: {Kamel.benaissa | Daniel.diep}@ema.fr

KEYWORDS:

Manufacturing planning and scheduling, multi-agent systems, agile manufacturing systems.

ABSTRACT

For manufacturing industries today, strategies for maximising the responsiveness of manufacturing systems are expressed by the process of production scheduling. This paper introduces a multi-agent based approach for process planning and scheduling. The general framework of an agent based approach for agile manufacturing systems scheduling is presented. Simple rules have been taken to construct a methodology for the development of negotiation mechanism in agent-based manufacturing systems scheduling.

INTRODUCTION

As the world market is becoming increasingly dynamic and turbulent, manufacturing enterprises require reactive manufacturing systems that are capable of responding rapidly to dynamic changes, to fulfil customer needs. It is essential for new manufacturing control strategies to be adopted, to maximise the responsiveness of manufacturing systems. In response to the new strategies, the integration of process planning and production scheduling will be required.

This paper introduces a framework which integrates process planning and production scheduling, as a means to achieve agile manufacturing and to minimise the production problems as mentioned above. This framework is developed based upon the concept of multi-agent systems (MAS), because the management of the constraints of reactivity passes by local treatments of tasks and through a process of communication assuring a global optimal solution (a sub-optimal solution can be also considered).

Section 2 briefly describes the context of distributed manufacturing process planning systems and related works. Section 3 presents the European project for which is developed this work. Section 4 presents the concept of the approach of planning and the underlying principles. A case of study in Section 5 illustrates the solving of a typical planning problem. Perspectives and conclusions are discussed finally.

AGENT IN MANUFACTURING SYSTEMS

Multi-agent systems have been recently proposed as a promising technology and an appropriate way to deal with the new trends and challenges in manufacturing systems, and to offer a convenient environment for designing and implementing distributed manufacturing solutions. Some examples are given by METAMORPH [1], CIIMPLEX [2], or PABADIS [3].

In multi-agent manufacturing systems, agents may represent single resources (a work cell, a machine, a tool), workers involved in a process, etc. as well as products, customers or providers. All together, they perform individual tasks, in interaction between them, in order to fulfil production functions (such as procurement, local planning, task assignment, scheduling, execution control, dispatching, etc.).

Scheduling issues

Scheduling is an optimization process where limited resources are allocated over time among both parallel and sequential activities. It is a difficult and complex problem solving, particularly when it takes place in an open, dynamic environment. Scheduling problems are dynamic in nature, provided that new orders are received continuously, while operations and complete orders are getting finished. Duration and service time may not be known precisely, machines may become unavailable and additional resources can be introduced at any time in the system. The starting time and the processing time of a task are also subject to variations; actually, a task can take more time than estimated or less time than anticipated, and tasks may arrive earlier or later. Because of its highly combinatorial aspects and its practical interest for manufacturing systems, the scheduling problem has been widely studied in the literature through various alternative methods and technics, such as constraint propagation techniques, simulated annealing, taboo search, genetic algorithms, neural networks, etc. [4],[5]. Due to the distributed aspect and parallelization characteristics of the methods, multi-agent technology has recently been considered as a successful attempt to resolve this problem [6].

Related works have demonstrated the interest of negotiation protocols between agents in order to reach an optimal scheduling solution. Considering rational agents under a game theory formalism, it is shown that agent co-operation tend to nearly optimal solutions [7], [8]. However this theoretical viewpoint is not easily transferable to industrial cases.

PABADIS'PROMISE PROJET

The PABADIS'PROMISE project[1] addresses flexible manufacturing systems submitted to highly varied demands in quantity (from one-of-a-kind series to batches) as well as in production orders (customized production) [9].

PABADIS'PROMISE architecture

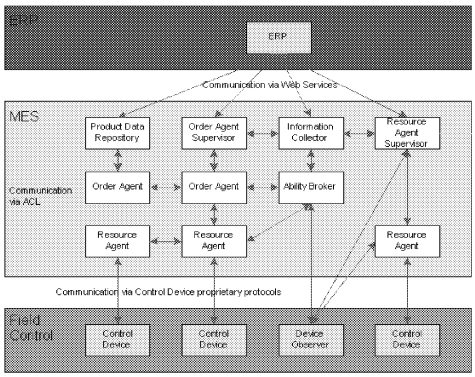


Figure 1 PABADIS' PROMISE architecture

PABADIS'PROMISE covers the three main levels reflecting the automation pyramid structure [9,10]: Enterprise Resource Planning (ERP) – Manufacturing Execution System (MES) – Field level (Figure 1).

ERP in PABADIS'PROMISE is a conventional ERP system that provides the production orders given by the users to the MES system and represents customers of the system. At the middle level, the ERP system is represented by the Order Agent Supervisor, which is an interface, responsible for communication between the ERP and the MAS.

MES in PABADIS'PROMISE represented by the MAS performs the MES functionalities and consists of two main types of agents: Order Agents (OA), representing the customers (orders) and the Resource Agents (RA) providing production abilities. The abilities of the RAs are maintained by the Ability Broker that manages a database of abilities and provides brokering functionality to the OAs.

And finally, *the field level* is a distributed environment, controlled by the RAs and offered to the MAS as a set of "abilities" provided by the field level to the customers of the system and represented by resources. A resource can be a robot, a conveyer line or even the whole plant.

A scheduling mechanism is currently under development (resource-oriented), initialized when a production order comes from the ERP. This paper addresses fist the decomposition in sub-orders and the decomposition of the Bill of Materials (BOM), which are required to start a production order.

DECOMPOSITION OF PRODUCTION ORDERS

An approach has been used by David He [11] to solve the problem of scheduling. A manufacturing system consists of types of operation: machining and assembly. A *digraph* represents a sequence of operations. In this graph, a node represents either the production of a part (P), or an assembly (A). The figure 2 illustrates the associated digraph for a simple sequence of machining and assembly.

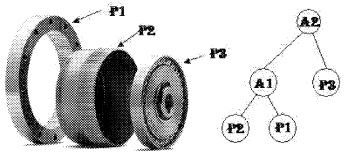


Figure 2 Digraph decomposition

A digraph can be decomposed in simple digraphs, i.e. graphs where there is at most one assembly operation at each level. And considering this decomposition, one agent is assigned to one simple digraph.

The planning algorithm

Inspired by the same author, we propose an algorithm which aims at minimising the overall completion time.

The OAS takes the role of *Schedule Manager* and organises the building of the overall scheduling and takes the main decisions. OAS consults the Product Data Repository, and creates the Order agents, associating each of them to one simple digraph.

The following notations must be introduced:

- *SAJ*=set of schedulable jobs, i.e., set of the jobs that can be scheduled because all their preceding jobs are scheduled, or they have no preceding job;
- *NSJ*=set of non-schedulable jobs, i.e., set of the jobs that cannot be scheduled because some of their preceding jobs have not been scheduled yet;
- *SJ*=set of scheduled jobs.

Among the OAs with schedulable jobs a leader is designated (in our actual algorithm, leader will be the one having the shortest makespan).

Once the Leader OA is identified, he combines his own schedule with those from the other OAs, calculating a temporary global schedule, and then, going to request negotiation with the others for proposing a better solution (a shortest makespan).

The Global algorithm is presented as following:

1. OAS receives the production Order form the ERP, and analyse the process sequences found in the PDR.
2. OAS realises the Digraph decomposition, and OA creation and assignation.
3. Each OA defines his tasks in one of the sets (SAJ, NSJ or SJ).
4. OAs with *SAJ* task requests the abilities from the AB.
5. OAs calculate local schedule.
6. OA Temporary Leader is designated.
7. Formulation of temporary global schedule by the leader.
8. Temporary leader asks the Others OAs for Negotiation.
9. If an other OA is able to propose changes (only with his initial tasks) to the proposed makespan,

[1] PABADIS'PROMISE (PABADIS based Product Oriented Manufacturing Systems for Re-Configurable Enterprises) is a STREP project funded by the EC and referenced FP6-IST016649. It refers to the former FP5 project PABADIS (Plant Automation Based on Distributed Systems).

making it shortest, he becomes the new temporary leader.

10. Back to step 7 if new leader.
11. Back to step 4 if no negotiation, or end the algorithm if all tasks are scheduled.

SOME RESULTS

To show the effectiveness of the proposed agent-based scheduling methodology in terms of solution quality computational experiments were conducted. We chose the example of a system consisting of two machines and one assembly station.

We decide to run agent based approach with a complex sequence on machining and assembly, in order to test the coherence of the algorithm and its efficiency. The chosen digraph is illustrated in figure 3, and execution times for each task were reported in Table 1.

Part	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Time (mn)	6	8	7	2	5	5	12	18	9
Assembly	A_1	A_2	A_3	A_4	A_5	A_6	--	--	--
Time (mn)	9	5	3	5	7	7	--	--	--

Table1 Tasks execution times

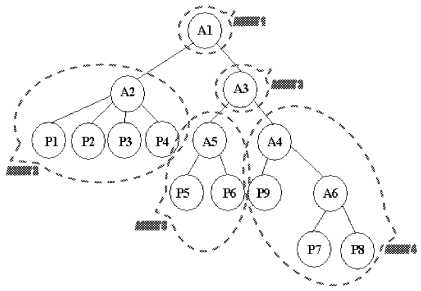


Figure 3 Machining Sequence

Developed under Java, using the Jade platform [11], the simulator, in which inputs are the complete digraph and the number and type of resource machines. As an output, we recover the Gantt chart carried out for each step by each agent, and thus, the Gantt chart representing final planning.

Some outputs of a simulation run are presented under the shape of the Gantt charts of Figure 4.

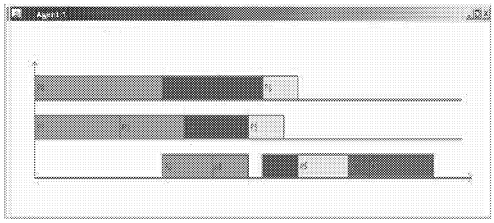


Figure 4 Snapshots of Gantt charts

This simple case implements the decomposition of an order into a set of agents and shows how the individual schedulings are merged after negotiation in an optimal scheduling.

The objective is using these results for the evaluation of the convergence from the level of local schedules of distributed agents towards a total schedule of the assembly sequence.

CONCLUSION AND FUTURE WORKS

To demonstrate the effectiveness of the proposed methodology, computational experiments were conducted. The results of tested problems show that the scheduling method obtains optimal or near optimal solutions.

The simulation results are also useful for a comparison with other research methods of scheduling problems (Taboo search, genetic algorithm...).

However, the static aspect of the suggested approach obliges us to think about a dynamic adaptation and a real time management of the production process. Ongoing work of developing scenarios and application cases within the PABADIS'PROMISE project are currently under definition and formalisation; these will be used for test implementation and evaluation of our scheduling algorithm.

REFERENCES

[1] Maturana F., Shen W., Norrie D.H. METAMORPH : an Adaptive Agent-based Architecture for Intelligent Manufacturing. Int. Journal of Production Research. 1999 vol 37 n°10, pp. 2159-2173

[2] Peng Y., Finin T.W., Cost R.S., Chu B., Long J., Tolone W.J., Boughannam A.: Agent-based Approach for Maunfacturing Integration: the CIIMPLEX Experience. Applied Artificial Intelligence, 1999 vol 13 .

[3] Klostermeyer A., Klemm E.: PABADIS – an Agent based Flexible Manufacturing Concept. Proc. IEEE Int. Conf. on Industrial Informatics (INDIN), Aug. 2003.

[4] P. Massotte, "Analysis and approaches for the management of com-plex production systems", in The Planning and Scheduling of Pro-duction Systems, Edited by A.Artiba and S. Elmaghraby, Chapman & Hall – 1997.

[5] M. Zweben, M. S. Fox, Intelligent Scheduling. Morgan Kaufmann Publishers, San Francisco, CA, 1994.

[6] A. D. Baker, "A survey of factory control algorithms which can be implemented in a multi-agent heterarchy: dispatching, scheduling, and pull", in Journal of Manufacturing Systems, Vol. 17, N°4, 1998, pp. 297-320.

[7] J. Reaidy, P. Massotte, D. Diep, "Comparison of negotiation protocols in dynamic agent-based manufacturing systems", in Proceedings of IEPM'03, Int. Conf. on Industrial Engineering and Production Management, 2003, Porto, Portugal.

[8] D. Diep, J. Reaidy : "Scheduling Agents in a Distributed Flexible Manufacturing System", in Proceedings of ISIE 04, IEEE Int. Conf. on Industrial Electronics, Ajaccio, France, May 4th-7th, 2004.

[9] A. Lüder, J. Peschke, A. Bratukhin, A. Treytl, A. Kalogeras, J. Gialelis, "The PABADIS'PROMISE Architecture", in Proceedings of the International Congress ANIPLA, 2006

[11] He D., Babayan A., Agent based agile manufacturing system, Int Journal of Automation, 1997.

[10] A. Bratukhin, B. A. Kahn and A. Treytl "Resource-oriented scheduling in the distributed production". in Proceedings of the International Congress ANIPLA, 2007.

[11] JADE, Java Agent Development Framework, 2005, available at http://jade.cselt.it/.

GLOBAL MULTI-PLANT FLEXIBLE NETWORKS SIMULATOR

Eduardo Castellano, Iñaki Zugasti, Jone Uribetxeberria and Juan Manuel Besga
IKERLAN Technological Research Center
Paseo J. M. Arizmendiarrieta, nº2
Arrasate-Mondragón, 20500
Spain
Email: ecastellano@ikerlan.es

KEYWORDS
Manufacturing, Network, Decision support systems, Hybrid simulation.

ABSTRACT

A key issue faced by organisations today is the challenge posed by how to deliver the different products demanded by customers, in different markets, at any given moment in time, and preferably individually customized, as cheaply and as quickly as possible. The ongoing research presented in this paper attempts to help globalised organisations to identify alternative multi-plant network configurations, and management strategies, to respond, through a mass customisation business strategy, to different demand requirement scenarios, within required cost and time restrictions. In order to facilitate the understanding of the consequences of adding or removing operational units, processes, reallocation of resources, etc, over the global network performance, a computational decision support system approach has been taken.

INTRODUCTION

Market globalisation, world-wide procurement, geographically distributed plants, more sophisticated customer requirements, increases in product variety, the rapid entry of new technologies and greater uncertainty due to the reduction in product life cycles, the emergence of new markets, etc..., are hardening global competition in general, creating a new dynamic environment for supply networks. This new dynamic environment has led to the development of new managerial approaches, due mainly to the transformation of competitive forces, with the appearance of responsiveness to demands, with a greater degree of customisation, as a key factor (Kidd 1994).

Current production systems require the capability to change their performance dimensions in terms of range -total number of states that the system is capable of achieving- and response -the ease with which changes can be conducted within the range-. Also, current productive systems must face these new challenges maintaining in turn high levels of performance in traditional competitive aspects such as cost. In the words of Vokurka and Fliedner (1998), what is needed is: *"the ability to produce and market successfully a broad range of low cost, high quality products with short lead times in varying lot sizes, which provide enhanced value to individual customers through customisation"*. All of which has led to the need for moving to a multi-focus strategy, in which the concept of flexibility, as the ability to respond adequately to change in different environments, takes on special relevance. The business strategy of mass customisation tackles this issue through the design of productive systems in which product differentiation is conducted as late as possible. The main idea is that one-of-a-kind type products are manufactured with high levels of quality and fast delivery, with the low costs of mass production (Anderson and Pine II 1997). This strategy achieves, therefore, a compromise between the advantages of product customisation -economies of scope-, guaranteeing fast response times for customized demands, with productivity and low costs associated with economies of scale, through decoupling points (Tseng and Jiao 1998).

The set of changes described above affect the way in which we conceive manufacturing and distribution networks, both on a configuration level, management level, and performance. Globalised markets, for diverse reasons, push networks to globalisation, i.e. the establishment of new plant nearer to local markets to reduce logistic costs and improve responsiveness, the establishment of production plants in countries with lower labour costs, etc (Porter 1986). The existence of network constituent elements in different geographical locations, in different countries, likewise increase complexity and uncertainty when it comes to managing it (Bhatnagara and Sohal 2005).

In order to deepen understanding of the consequences of multi-plant network design, and strategic policies, over the global network performance, a computational decision support system approach has been taken. The simulation approach will allow us to conduct, for a multitude of scenarios and conditions, systematic testing of the structure and operation of this type of networks, its behavioural patterns and properties, in order to identify, both alternative flexible multi-plant network structures, as well as those strategies, policies, and rules, that are most adequate for their management, both on a local and a global network level, at low cost and risk (Shapiro 2001).

And thus, approach the challenge of this research, about how to enable a globalised organisation identify, based on demand orders and following a mass customisation business approach, alternative spontaneous multi-plant network configurations, and their management strategies, to facilitate respond different customized demand scenarios, in the most efficient, cost and time, possible manner.

The outline of this paper is organised as follows. Section 2 presents some specific demand driven supply networks concepts, which place in context the conceptual model developed. In section 3, the research approach is specified. Section 4 describes the conceptual model developed, based on the heuristics elicited from a case study experts. In Section 5, some views of the conceptual model implementation in the simulation environment are presented. Finally, section 6 shows some preliminary conclusions and the future steps of this on-going research.

DEMAND DRIVEN NETWORKS CONCEPTS

Depending on the characteristics of product orders, i.e. volume, number of variants, uncertainty in demand, product life cycle length, lead-time accepted, etc, there are different models of supply network (Fisher 1997), different types of manufacturing process (Hayes and Wheelwright 1979), as well as planning and control systems (Berry and Hill 1992), that are more adequate. These three "models", are very significant as background to place in context the development of the conceptual model of this research. We next move on to describe briefly each of them, along with a final note about the decoupling point concept, which allows us to relate the customer orders characteristics, and operations management, to the mass customisation business strategy adopted.

Customer Orders Characteristics and Supply Network Structure

Regarding the relationship between the characteristics of orders and supply network models, Fisher (1997), based on the cases of Campbell Soup and Sport Obermeyer, distinguishes between functional products with foreseeable demand, and innovative products with unpredictable demand. For those belonging to the first group, the author assigns a physically efficient type of supply network, the aim of which is to maximise efficiency at the lowest cost possible, high levels of manufacturing level resources, a strategy of inventory minimisation and a reduction in lead times. As regards the second group, innovative products, Fisher recommends market responsive type supply networks. The aim of these ones is to respond quickly to a demand with a high degree of uncertainty for minimising stock-outs and obsolete inventories. For this, excess manufacturing capacity is required, as well as product parts and finished products broad buffers, aggressive investment in lead time reduction, supplier selection based on their speed, quality and flexibility, and a modular type design strategy which facilitates the postponement of customisation of the product as late as possible.

Customer Orders Characteristics and Processes

As regards the relationship between product order characteristics, and manufacturing process type, the product-process matrix (Hayes and Wheelwright 1979) establishes two axes. The first one contains product characteristics, from products with a high number of variations and low volumes of orders (unique and one-of-a-kind products), to products with a low number of variations and high volume of orders (standard products). The second axis in the matrix, which refers to manufacturing processes, assigns, to each one of the previous types, manufacturing process of the type; project manufacturing for unique products, job shop type processes for one-of-a-kind products, flow shop for products with many variants and low volume, line flow for few major products with high volume (i.e. automobile), and finally, continuous manufacturing processes for standard products with high volumes. This model has been empirically validated in different empirical works.

Customer Orders Characteristics and Operations Management

On the management level, manufacturing operations, inventories and transport at each production unit in the supply chain are planned on different hierarchical levels. Depending on the level of aggregation, and planning horizon, can be distinguished: Sales and operations planning; Master scheduling; Materials planning, and; Production activity control. As a function of market order characteristics, different approximations may be adopted at each of these hierarchical management levels (Berry and Hill 1992; Selldin 2003).

Table 1: Order characteristics and management strategies relationship (Selldin 2003)

Product Orders Characteristics			
Product type Product range Individual Product volume per period	Standard Narrow High		Customised Wide Low
Management Strategies			
Sales and operations planning level	Level strategy... when via inventories or the backlog of customer orders a stable rate of production can be maintained		Chase strategy... when production follows fluctuations in demand
Master scheduling level	Make-to-stock (MTS)... when production is based on forecasts	Assemble-to-order (ATO)... build-to-order, configure-to-order, and finish-to-order	Make-to-order (MTO)... when production is based on customer orders
Materials planning level	Rate-based planning... when a production rate is set and harmonised through the manufacturing processes.		Time-phased planning... when the different activities in the manufacturing processes are planned to start at different points in time depending on the product structure, production lead-times and due-date.
Production activity control level	Just-in-time type (JIT)... when shortages of material in the manufacturing process triggers a signal for just-in-time replenishment		Material requirement planning type (MRP)... when MRP is often used to plan the activities based on a time-phased planning

Decoupling points and flexibility

The dividing line between orders manufactured based on forecasts (MTS) and those that are assembled, manufactured, or designed, according to customer orders (ATO, MTO, ETO), determines a very important point in manufacturing processes and supply networks, known as the Customer Order Decoupling Point – CODP (Mason-Jones et al. 2000). Upstream the CODP there are few product variants and volumes are high, downstream the

CODP each product may be unique and customised for a specific customer. Therefore, the CODP marks the point at which an order becomes attached to a specific customer, conditioning therefore the capacity of a supply network to provide different grades of customised product, at a particular cost and delivery schedule.

In a multi-plant network there may be multiple CODP, i.e. within the same operation unit certain products may be manufactured MTS and others MTO, or even in between plants. For instance, to achieve supply chain mix flexibility, the capacity of the manufacturing system to alter the relative production amount between products in a product mix, both, a multi-product plant strategy, as well as a certain number of single-product plants strategy may be adopted. In this last case, each single-product plant would have a low grade of mix flexibility being able to develop high productivity ratios, and would at the same time form part of a superior structure and strategy leading to mix flexibility on a firm level (Vereecke and Van Dierdonk 1999). The multi-plant option, multi-plant networks, is very interesting for creating flexible production capacity, transferring production volume from overloaded plants to others that at that moment in time have excess capacity. The potential configurations of multi-plant networks can become, therefore, a key aspect in obtaining, via very different means, the levels of flexibility desired on a firm level. Thereby, for each case being analysed one must consider what type of flexibility is desired for each unit of analysis, and define the particular manner in which it shall be developed to obtain the objectives as efficiently as possible for the system as a whole.

RESEARCH APPROACH

The approach adopted can be classified as an engineering approach for performance enhancement of systems (Pritsker 1997). For developing the research we have based on: (1) Previous studies identified in the literature; (2) A case study research of a bicycle manufacturer, denoted in this paper as GBIKE, and; (3) A simulation environment. It is a combination therefore of case study research and simulation modelling research. Case studies are frequently used for exploratory and theory building research (Yin 1994). The selected case study, GBIKE, is a bicycle manufacturer with a global multi-plant network and urgent requirements for introducing flexibility in their response to markets demanding products with a high degree of customisation. The simulation modelling approach is also fundamental to this research due to the high levels of interdependencies between constituent elements of these multi-plant networks, their inherent feedback loops, non-linearities, and delays, that make network behaviour in the face of market demand variations a dynamic process which generally produces counterintuitive behaviours over time (Sterman 2000), and therefore makes purely analytical approaches to the problem inadequate (Fowler 1998).

The modelling and simulation methodological framework followed in the research is the one of systems dynamics - SD, as its validity has been demonstrated for the design of company simulations and strategic decision-making laboratories (Sterman 2000). Its main steps, in outline, are the following ones (Coyle 1977; Sterman 2000):

1. Elicitate the conceptual model.
2. Implement the conceptual model in the simulation environment.
3. Calibrate and validate the simulated model based on empirical data.
4. Conduct structural and behavioural analysis of the simulated model under different conditions (i.e. policies, operations...), and scenarios (i.e. customer demand scenarios).
5. Make managerial suggestions based on the knowledge of the model behaviour to improve the real world system performance.

As regards the identified stages, section 4 presents the conceptual model corresponding to stage 1 of the methodology. And section 5 shows different views of the simulator developed based on the conceptual model. Stage 3, right now, is under development.

DEMAND DRIVEN MULTI-PLANT NETWORK CONCEPTUAL MODEL

This section collects the results from the conceptual model elicitation for GBIKE. Interviews and meetings have been held with experts at GBIKE, mainly following Group Model Building methodology (Vennix 1996). The result of these sessions has been completed by literature studies related to the problem.

Building Blocks Relationships

The conceptual model associated with the multi-plant network developed to provide mass customisation environments, has been structured into five conceptual blocks, as illustrated in Figure 1.

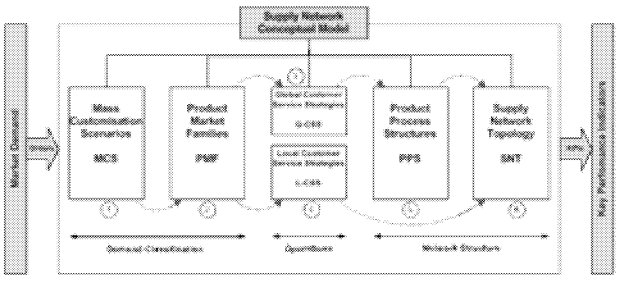


Figure 1: Supply network conceptual model diagram

As demand orders arrive they classified according to the customization required (MCS), and the product family to which they belong and the market to which they are directed (PMF). As a result of this classification, orders are associated with one or more strategies by which they shall be served to customers (CSS). These strategies can be of two types: (i) *Global*, whereby it is specified, in product and process structures (PPS), what components are to be manufactured, subcontracted or purchased, and the form of

the network topology (SNT) that will supply the product; (ii) *Local*, whereby the restocking procedure is established for each network node. A set of key performance indicators are obtained for measuring the performance of the system order transmission process: Inventory level; Compliance with delivery times; Use of resources, etc. Each block is described in greater detail below.

Building Blocks Description

Mass Customisation Scenarios (MCS)

In this block, five scenarios have been established for the different product customisation levels offered. Taking into account factors such as stock availability of the finished product, habitual order volumes, or delivery schedule requested, each MCSs tends to have a different combination of such factors. Table 2 presents habitual values for the cited factors, and considers separately, initial orders of those which are made to respond to consumption. Additionally, habitual sales types for each MCS are included.

Table 2: MCSs factors values for a bicycle manufacturer

MCS	Initial Order			Replenishment Order			SPT
	Stock	Order Size	Delivery Time	Stock	Order Size	Delivery Time	
1. Standard	Yes	High	Delivery Schedule	Yes	Medium	Short	Shopping Centres Specialized Superstores Specialized Shops
2. Menu	No	Low	Short	No			Specialized Shops
3. Selection	No	High	Delivery Schedule	No	Low	Short	Specialized Superstores
4. Adaptation	No	High	Delivery Schedule	No	Medium-Low	Short	Specialized Superstores
5. Pure Customisation	No	Low	Short	No			Specialized Shops

Product-Market Families (PMF)

This block allows to define the competitive position of the company in different markets (M_1, M_2, ..., M_m, in Figure 2), for different product families (PF_1, PF_2, ..., PF_p, in Figure 2), which determine the offer.

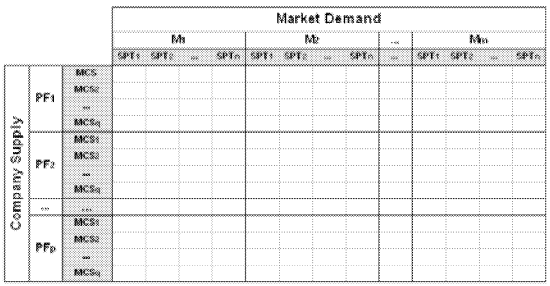


Figure 2: Product Market Family (PMF) matrix

The level of detail is increased by specifying, for each market M, the different types of sales points available (SPT_1, SPT_2, ..., SPT_n, in Figure 2), i.e. shopping centres, super-mega-stores or specialist shops, and for each product family PF, the different levels of product customisation (MCS_1, MCS_2, ..., MCS_q, in Table 2) that they will offer to their customers, which has been described above.

For each cell of Figure 2, resulting from the overlap between M-SPT-PF-MCS, the company must establish the sales volume objectives, price and margin, delivery times and the strategies with which it will serve customers.

Customer Service Strategies (CSS)

This block is used to determine the strategy to serve customer orders according to the PMF to which they belong. Orders for standard products, or with a minimum customisation level, which can be fulfilled during distribution phases, can be served using a stocking, manufacturing and assembly strategy, to stock (MTS) - type 1 of Figure 3.

The greater the degree of customisation required in the product demanded by the customer (types 2 and 3 of Figure 3), greater the level of penetration in the supply network of the point at which manufacturing is associated with the customer CODP, limiting therefore the possibilities of producing to stock (MTS) certain components in the structure, and to order (MTO-ATO) the rest. In the borderline case (type 4 of Figure 3), stocking, manufacturing and assembly are conducted specifically by order (MTO-ATO).

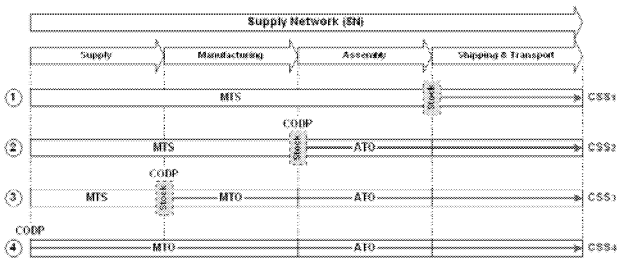


Figure 3: Customer Service Strategies (CSS)

The increase in the level of penetration, in the supply network, conditions the typical decisions associated with product delivery, such as component purchase, or subsets, for suppliers located in different regions or countries, or rather, the subcontracting of certain operations, or the assignment of production phases, to different locations of production plant. Type 4 orders, with short delivery times, may cause manufacturing and assembly to be performed in the same factory, which must be near the delivery point, with suppliers nearby who respond quickly to the specific characteristics of such orders. On the contrary, type 1 orders, with equally short delivery times, allow a greater number of multi-plant alternatives to be considered for supplying the demanded product.

In the conceptual model, this type of decisions are named as Global policies of the CSS (G-CSS), given that it affects the global configuration of the supply network. In addition to these policies, CSSs include Local policies that have less impact on the supply network, affecting the existing relationship between pairs of nodes, establishing the procedures by which the material orders made by the customer node are to be supplied from the supplier node (L-CSS).

Product-Process Structures (PPS)

This block represents the different alternatives for generating the product to be delivered to the customer, based on three basic options of process assignment: (i) Self manufacture-assembly; (ii) Purchase from supplier, or; (iii) Subcontract.

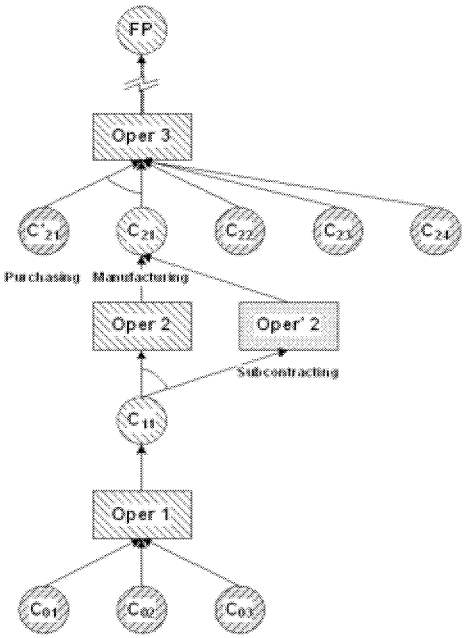


Figure 4: Product Process Structures (PPS) generation alternatives

The criteria used to conduct the assignment in each case constitute another of the elements that configure each CSS.

Supply Network Topology (SNT)

This block is used to define the basic elements for representing material and information flows that occur in the supply network. They can be of two types:

- *Nodes*: Basic elements of the topology representing the points of the supply network, where transformations or storage occur, to serve the product demanded by the customer. They are associated with a physical location.
- *Arcs*: Elements of union between two nodes. They indicate the transportation of materials (*component arcs*) and the flows of information (*information arcs*) between nodes.

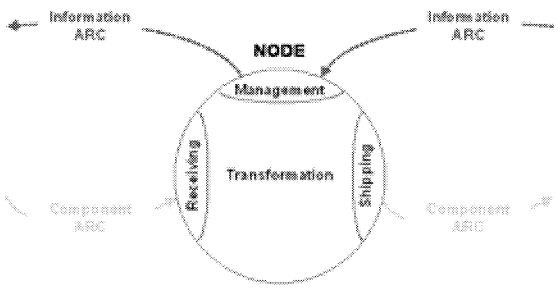


Figure 5: Node and Arcs general diagram

The SNT is formed from the set of nodes and arcs that may take different forms depending on the CSS established to serve orders and PPS flexibility.

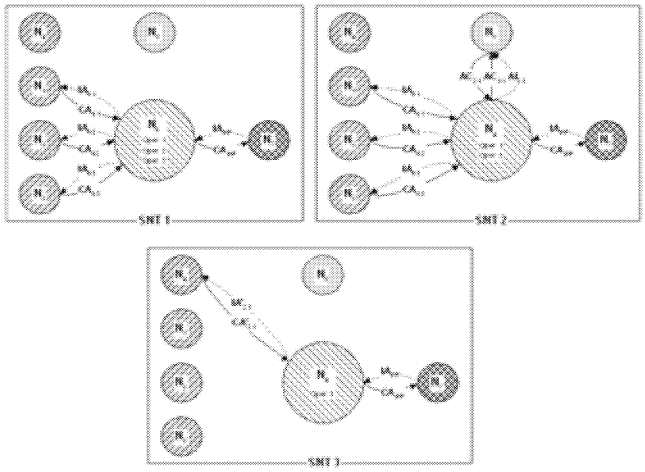


Figure 6: Example of SNT associated with different CSS

CONCEPTUAL MODEL SIMULATION

As mentioned previously, the high level of interdependence between the elements of these networks, their inherent feedback loops, non-linearities, and delays (Sterman 2000), mean that purely analytical approaches to the problem are not sufficient (Fowler 1998). Thus come out the need to attend simulation environments that have tested its validity for this purpose (Shapiro 2001). In the field of supply chain management simulation, both systems dynamics – SD (Towill 1996; Akkermans et al. 1999; etc…), as well as agent based modelling - ABM (Parunak et al. 1998; Swaminathan et al. 1998; etc…), and hybrid approaches of both (Akkermans 2001; Schieritz and Größler 2003; etc…), enclose one of the greatest amount of work in this area.

With respect to the implementation of the conceptual model in a simulation environment, a multi-paradigm SD-ABM approach has been used, via Anylogic simulation tool. This allows the different dimensions identified to be modeled in the detail, and simulated, as best suits to each one.

Following, some views of the simulator runs for a number of simple tests are shown. These tests are not based on real data but just want to illustrate some of the capabilities and functionalities that the simulator can easily perform.

Test 1 - Capacity utilization and replenishment strategies

Mass Customisation Scenarios - MCS…

- MCS1-Standard: FSX bicycle family, for Shopping Centres, Specialized Superstores, and Specialized Shops.

- MCS2-Menu (catalogue colours): Vertex bicycle family, for Specialized Superstores and Specialized Shops.
- MCS4-Adaptation (personalised colours): Vertex bicycle family, for Specialized Superstores and Specialized Shops.

Product-Market Families - PMF…

- FSX family for European markets, with normal distribution type function for the number of orders.
- Vertex family for European markets, with normal distribution type function for the number of orders and uniform type distribution functions for the colors and type of brakes.

Customer Service Strategies - CSS…

- G-CSS: The process activities Solder(of tubes to make the frame)-Painting(frames are painted)-Assembly(of all components) are done at Spanish plants with the options of MTO, ATO, and MTS.

- L-CSS: Replenishment of stocks of components and materials through order point.

Product-Process Structures - PPS…

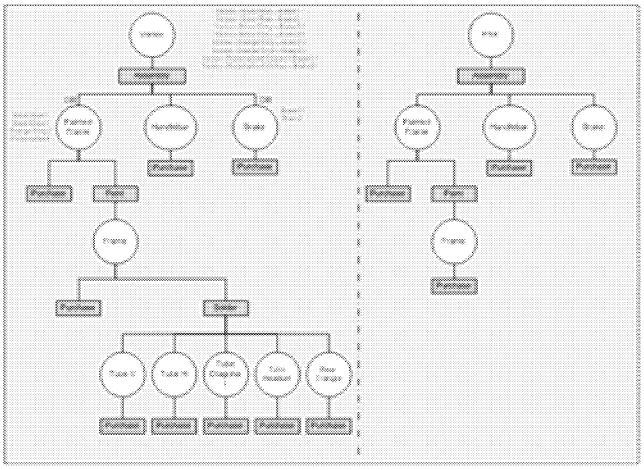


Figure 7: PPS for the simulation tests

Test 1 objective…

- To identify the main bottlenecks, capacity needs and capacity utilization, as well as the effects of different replenishment strategies (i.e. order point), over the network performance (cost, and delivery time of orders to the different sale points).

Simulation view in Figure 8…

- View of the activated nodes of the current GBIKE simplified version network during running time.
- Deviation of cost and delivery time from the order objective for Vertex and FSX orders.
- As an example, the stock level variations in time of a brake component for a certain plant following a order point L-CSS.

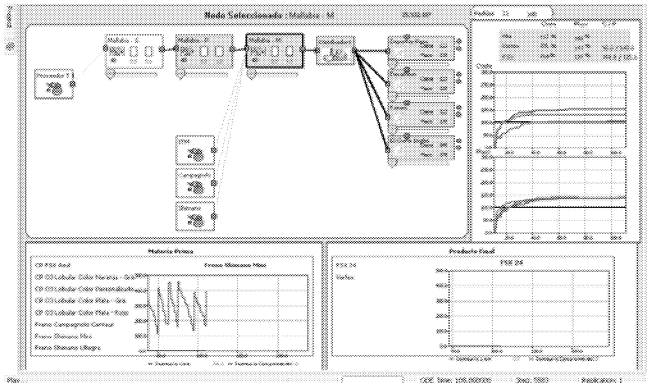


Figure 8: Test 1 simulation run time view

Test 2 – New markets and network topology

Mass Customisation Scenarios - MCS…

- As in Test 1

Product-Market Families - PMF…

- FSX family for European and Asian markets, with normal distribution type function for the number of orders.
- Vertex family only for European markets, with normal distribution type function for the number of orders and uniform type distribution functions for the colours and type of brakes.

Customer Service Strategies - CSS…

- G-CSS:

  For Vertex family, one G-CSS is tested:

  1. Process activities Solder(of tubes to make the frame)-Painting-Assembly(of all components) at Spanish plants, and MTO.

  For FSX family, four G-CSS are tested:

  1. For the Asian market: An assembly plant is placed at Turkey that works by ATO. The Turkish plant is supplied from Spanish Solder and Painting plants that work by MTO.
  2. For the Asian market: An assembly plant is placed at Turkey that works by ATO. The Turkish plant gets painted frames supplied from Spanish and local Turkish suppliers of painted frames that work by MTS.
  3. For the European market: The process activities Solder(of tubes to make the frame)-Painting(frames are painted)-Assembly(of all components) are done only at Spanish plants with the options of MTO, ATO.
  4. For the European market: Mix of G-CSS1 and G-CSS2, all FSX demand ATO through Turkish plant,

and all Vertex demand ATO and MTO from Spanish plants.

- L-CSS: Replenishment of stocks of components and materials through order point.

Product-Process Structures - PPS…

- As in Test 1

Test 2 objectives…

- Check several G-CSS for both families (Vertex and FSX) and markets (European and Asian) to see which of them achieve better results in terms of performance (time to market, cost).
- And identify the main bottlenecks, capacity needs and capacity utilization, as well as the effects of different replenishment strategies (i.e. order point), over the network performance (cost, and delivery time of orders to the different sale points).

Simulation view in Figure 9…

- View of the activated nodes, of an hypothetical GBIKE simplified version network with an FSX assembly plant in Turkey and local Turkish suppliers of painted frames, during running time.
- Deviation of cost and delivery time from the order objective for Vertex and FSX orders.
- As an example, the stock level variations in time of a brake component for a certain plant following a order point L-CSS.

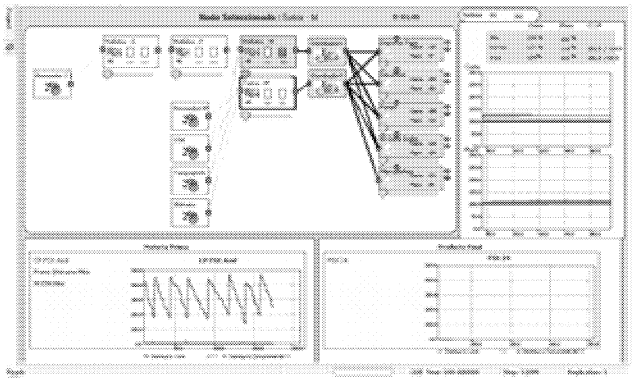


Figure 9: Test 2 simulation run time view

Test 3 - New markets and network topology optimization

Example of automatic optimization (cost, and delivery time of orders to the different sale points) of Test 2 G-CSS for both Vertex and FSX families, and European and Asian markets.

Optimization outputs therefore indicate the best G-CSS for each PMF of Test 2.

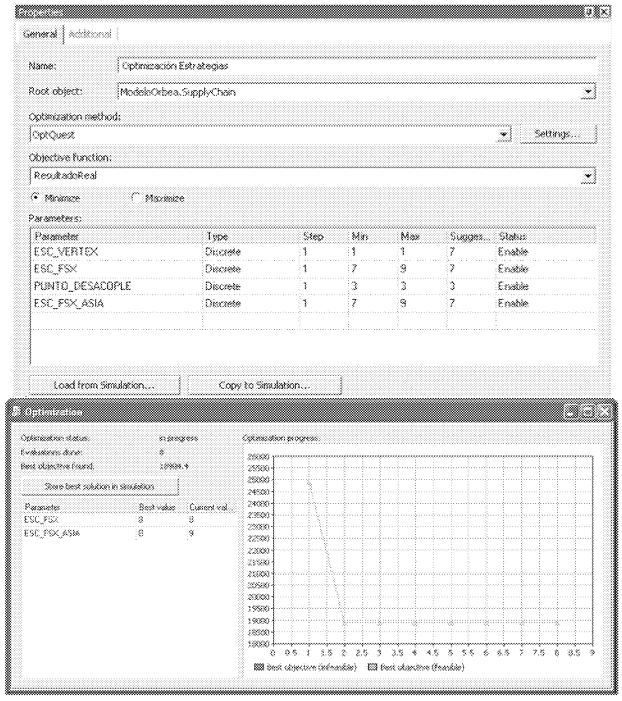


Figure 10: Test 3 optimization run time view

PRELIMINARY CONCLUSIONS AND FUTURE RESEARCH STEPS

An important aspect in obtaining adequate network flexibility and efficiency involves taking into account the precise moment at which customization occurs within the management process of the different customer orders. The conceptual model and simulator developed facilitates the definition of alternative mixed demand driven responsive and efficient multi-plant networks based on personalized customer orders. According to the experience of experts at GBIKE, the presented model and simulator can be easily adapted to many other companies which operate under similar environments. This subjective opinion awaits confirmation via new case studies.

As regards current and future research steps, the "Calibration and validation of the model behaviour based on real historical data from GBIKE" stage is right now under development. Once finished, it will allow us, in the short term, to cover the:

- Evaluation of GBIKE model behaviour under different conditions (i.e. policies, operations...), and scenarios (i.e. customer demand scenarios);
- Identification of efficient global flexible multi-plant designs and management strategies, for GBIKE potential customer demand scenarios.

And therefore, use this knowledge to develop support guidelines to help GBIKE to dynamically reconstruct, in practice, its global supply network through demand driven responsive and efficient multi-plant networks.

REFERENCES

Akkermans, H. 2001. "Emergent Supply Networks: System Dynamics Simulation of Adaptative Supply Agents". In *Proceedings of the 2001 Hawaii International Conference on System Sciences* (Island of Maui, HW, Jan.3-6).

Akkermans, H.; P. Bogerd; and B. Vos. 1999. "Virtuous and vicious cycles on the road towards international supply chain management." *International Journal of Operations & Production Management*, Vol.19 (5/6), 565-582.

Anderson, D.M. and B.J. Pine II. 1997. *Agile Product Development for Mass Customization, Niche Markets, JIT, Build-to-Order, and Flexible Manufacturing*. Burr Ridge, IL, Irwin/McGraw-Hill Professional Publishing.

Berry, W.L. and T. Hill. 1992. "Linking systems to strategy." *International Journal of Operations and Production Management*, Vol.12 (10), 3-15.

Bhatnagara, R. and A.S. Sohal. 2005. "Supply chain competitiveness: measuring the impact of location factors, uncertainty and manufacturing practices." *Technovation*, Vol. 25, 443-456.

Coyle, R. 1977. *Management Systems Dynamics.* Wiley, NY.

Fisher, M.L. 1997. "What is the right supply chain for your product?." *Harvard Business Review*, Vol. 75 (2), 105-16.

Fowler, A. 1998. "Operations management and systemic modeling as a framework for BPR." *International Journal of Operations and Production Management*, Vol.18 (9/10), 1028-1056.

Hayes, R.H. and S.C. Wheelwright. 1979. "Link manufacturing processes and product life cycles." *Harvard Business Review*, Vol. 57 (1), 133-140.

Kidd, P.T. 1994. *Agile Manufacturing – Forging New Frontiers*. Wokingham: Addison-Wesley.

Mason-Jones, R.; B. Naylor; and D.R. Towill. 2000. "Lean, agile or leagile? Matching yor supply chain to the marketplace." *International Journal of Production Research*, Vol. 38 (17), 4061-70.

Parunak H.V.D; R. Savit; R.L. Riolo; and S.J. Clark. 1998. "DASCh: Dynamical Analysis of Supply Chains". Working Papers of CEC & University of Michigan.

Porter, M.E. (ed.). 1986. *Competition in Global Industries*, Harvard Business School Press, MA.

Pritsker, A.A. 1997. "Modeling in performance-enhancing processes." *Operations Research*, Vol.45 (6), 797–804.

Schieritz, N. and A. Größler. 2003. "Emergent Structures in Supply Chains - A Study Integrating Agent-Based and System Dynamics Modeling." In *Proceedings of the 2003 Hawaii International Conference on System Sciences* (Island of Hawaii, HW, Jan.6-9).

Selldin, E. 2005. *Supply chain design. Conceptual models and empirical analyses*, PROFIL 22, Ph.D. dissertation, Linköping Institute of Technology, Linköping, Sweden.

Shapiro, J. 2001. *Modeling the Supply Chain*. Duxbury Press, North Scituate, MA.

Sterman, J. D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World.* NY: McGraw-Hill Higher Education.

Swaminathan, J.M.; S.F. Smith; and N.M. Sadeh. 1998. "Modeling Supply Chain Dynamics: A Multiagent Approach", *Decision Sciences*, Vol. 29 (30), 607-32.

Towill, D.R. 1996. "Time compression and supply chain management - a guided tour." *Supply Chain Management*, Vol.1 (1), 15-27.

Tseng, M.M. and J. Jiao. 1998. "Concurrent design for mass customisation." *Business Process Management Journal*, Vol.4 (1), 10-24.

Vennix, J. 1996. *Group Model Building: Facilitating Team Learning Using System Dynamics*. Chichester: John Wiley & Sons.

Vereecke, A. and R. Van Dierdonk. 1999. *Design and Management of international Plant Networks: Research Report*. Academia Press, Gent, Belgium.

Vokurka, R.J. and G. Fliedner. 1998. "The journey toward agility", *Industrial Management & Data Systems*, Vol.4, 165-71.

Yin, R. (1994) *Case study research: Design and methods* (2nd ed.). Beverly Hills, CA: Sage Publishing.

MODELING STRATEGIC AND OPERATIONAL DECISION-MAKING – AN AGENT-BASED MODEL OF ELECTRICITY PRODUCERS

Emile Chappin, Gerard Dijkema, Koen van Dam and Zofia Lukszo
Faculty of Technology, Policy and Management
Delft University of Technology
Jaffalaan 5
2628 BX Delft, the Netherlands
E-mail: E.J.L.Chappin@TUDelft.NL

KEYWORDS

Agent-based modeling, agent architecture, ontology, electricity producers.

ABSTRACT

Essential steps in building agent-based models are to conceptualize, quantitatively representing and implementing the behavior of agents. A formal approach is presented to model the behavior of consumers, institutions or companies as rational decision-makers. Key element is a generic ontology for socio-technical systems and a framework for conceptualizing actors as agents with strategic, tactic and operational behavior. The approach was used to represent the behavior of electricity producers who must decide on investments, day-to-day operations and deal with CO_2 emission-rights. This exercise amply demonstrates the value of using a formal conceptual framework in the development of agent-based models.

INTRODUCTION

Infrastructures for transportation, telecommunication, natural gas, electricity, etc. are essential for a modern industrial society. Infrastructure networks are technologically advanced large scale systems, which development, structure and operation is decided upon by a myriad of actors. These infrastructures are *socio-technical systems*, which are defined as systems in which physical elements are embedded in a social network of actors and institutions. The electricity infrastructure system, for example, not only consists of the physical infrastructure for electricity production, transportation and distribution, but also spans electricity producers, network operators, a variety of consumers, financial institutions, knowledge and service providers. Its evolution and operation is determined by discrete events, decisions by incompletely informed (ir)rational, agents on continuously operating, interconnected systems. As a consequence, the dynamics of this and other infrastructure systems cannot be predicted from scientific and engineering principles alone. Indeed, management science uses psychology, sociology and economy.

We believe that combining knowledge from both engineering and management science is required to successfully explore the emergent behavior of these systems by simulation. Appropriate models for socio-technical systems must represent a great many external factors, apparatus, technologies or facilities and a spectrum of actors. The latter determine government policy, develop and interpret regulations, invest under uncertainty in a global, continental or regional market, respond to consumer demand and preferences, ensure proper operation, avoid congestion and so on.

Particularly agent-based models (ABM) appear suitable to explore and gain insight in the dynamics of socio-technical, systems: in a socio-technical system, a subsystem of actors and a technical subsystem interact on different levels of aggregation (Ottens *et al.*, 2006). ABM can capture highly nonlinear interactions, common in the problems addressed by industry and government (North and Macal, 2007) and have, for example, been applied to cases in the process industry (Aldea *et al.*, 2004). Agent-based simulation models can cope with discrete events, multi-level (disaggregate) decision-making, emergent markets (Dam *et al.*, 2007; Nikolic *et al.*, in print). More important, ABM offer a good way to include the social and behavioral aspects. An ABM model describes the behavior of parts of the system (Schieritz and Milling, 2003), the system behavior emerges from the behavior of the smaller autonomous parts (Dam and Lukszo, 2006). Therefore, ABM simulations can be used for explorative research with different scenarios.

There is a wide range of application domains for ABM, since agent-based modeling originates from different research domains, such as social sciences, mathematics and from artificial intelligence (Wooldridge and Jennings, 1995; Epstein and Axtell, 1996; Axelrod, 1997; Bussmann *et al.*, 1998; Jennings, 2000; Weiss, 2000; Borshchev and Filippov, 2006). The agent paradigm in agent-based research can be very conceptual and has to be operationalised to be applicable in simulation.

A modular and flexible environment to set-up and execute ABM simulations has been developed (Nikolic *et al.*, 2004; Nikolic *et al.*, 2006; Chappin and Dijkema, 2007b) using an ontology as its foundation (Dam *et al.*, 2006), enabling effective representation and implementation for a range of problems and domains. The remising 'simulation engine' connects several software tools – Protégé as knowledge acquisition tool, Eclipse for code development, Repast (North *et al.*, 2006) as agent based simulation package and SPSS for statistical analysis of results – and is suitable for doing experiments in this variety of domains.

In this paper we focus on the conceptualization and formalization of agent behavior with respect to strategic and

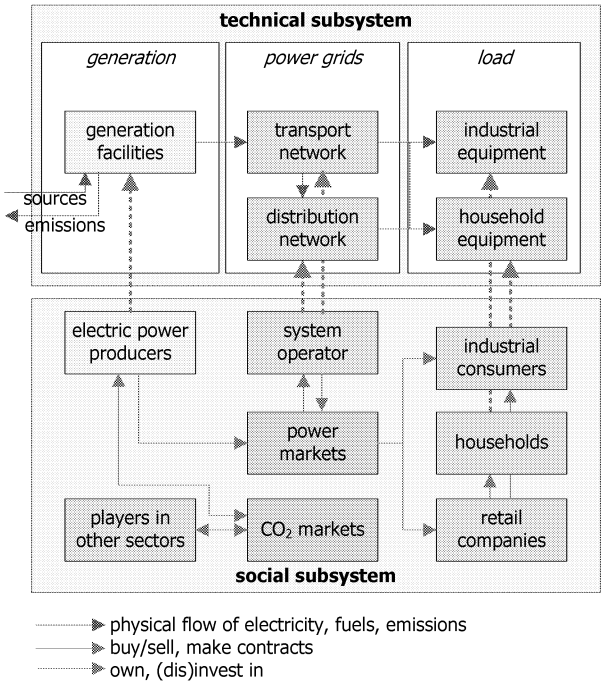


Figure 1. Socio-Technical Power Production System (Chappin and Dijkema, 2007b)

operational decisions. The case study presented addresses part of the electricity infrastructure, notably the electricity production sector where producers must deal with CO_2 emission-rights trading. The main objective of the case study was to obtain insight in the effect of the policy instrument CO_2 emission-trading (CET) on the types of power plants power producers prefer to build and the power generation portfolio that emerges from their decisions over time.

The main question addressed in this paper is: **How can the decision-making process of electricity producers, investing in and in control of power plants, be modeled for use in an agent-based model?**

The remainder of the paper is structured as follows: First the system description and the background of the case are given. Second, a generic conceptual framework for socio-technical systems consisting of agents and physical installations is presented. Third, the decision-making process of electricity producers is operationalised using the presented framework. Some results are given to illustrate the advantage of the approach used. Conclusions are given and some areas for future work are identified.

SYSTEM DESCRIPTION

The system studied is that wherein electricity producers play a pivotal role: the electricity infrastructure. Today, electricity producers must effectively operate in power-exchange markets, but also in markets for fuels, capital and emission rights. The electricity infrastructure is a socio-technical system: The social network is composed of power production companies, retail companies and consumers that trade on different markets. Governments and regulatory bodies are also part of this network. Power generation facilities, power grids and consumers equipment form the technical network. Figure 1 shows the interdependencies of the social and technical networks. Electricity producers thus must operate and invest in power plants respecting current rules and regulations. Consumers invest and operate their end-user equipment. Suitable power grids operated by distribution companies and/or controlled by government connect the two. Each of these must anticipate and act upon demand, market and regulatory developments expected in interdependent social and technical subsystems.

CET affects producers decisions to invest by forcing the possession of permits for emitting CO_2 (Laurikka and Koljonen, 2006; Olsina et al., 2006). The amount of permits issued is limited and can be traded amongst parties on emission markets (Svendsen and Vesterdal, 2003; Vesterdal and Svendsen, 2004; Schleich et al., 2006). CET is implemented because it is assumed that it would lead to the most cost effective emission reduction due to the "invisible hand" (Smith, 1776; Svendsen, 1999). Because the long term impact of CET is unknown and serious experience is lacking, we conjectured that agent-based simulations could help to provide insights (Chappin and Dijkema, 2007a).

The main impact of CET on total sector CO_2 emissions would emerge from altered investment decisions of power producers (Chappin and Dijkema, 2007a). Governments implemented this instrument because it is assumed that it will lead to a less CO_2-intensive generation-portfolio. However, the producers, the agents in the electricity infrastructures, are autonomous. History shows that individual producers do not exhibit the same decision-behavior. Furthermore, investment and disinvestment decisions are discrete events about capital-intensive pieces of equipment. Agent-based models are suitable for explicitly simulating this. Our research into CET effects is of an exploratory nature, mainly because of the lack of historic data on emission-trading and its impact. Exploratory modeling for decision-support is very well possible with agent-based modeling.

CONCEPTUAL FRAMEWORK FOR SOCIO-TECHNICAL SYSTEMS CONSISTING OF AGENTS AND PHYSICAL INSTALLATIONS

The conceptual framework presented here consists of the definition of an ontology, which is a formal specification of concepts from a certain domain. Here the domain is that of socio-technical systems and the ontology aims at being generic so that different application domains can be expressed with it. The concepts defined in the ontology form the communication language of the agents as well as the framework in which the decisions of the agents is formalized. The concepts are abstract descriptions of classes.

Figure 2 shows a fragment of the ontology that describes the agents and physical installations as nodes. Agents in this model own physical apparatus that are on the level of physical assets. Physical assets do not act themselves as they are passive and controlled by agents owning them. The main concepts from Figure 2 are introduced below:

- **Agent:** Representation of an actor or institution. Can be the owner/controller of physical installations. Agents

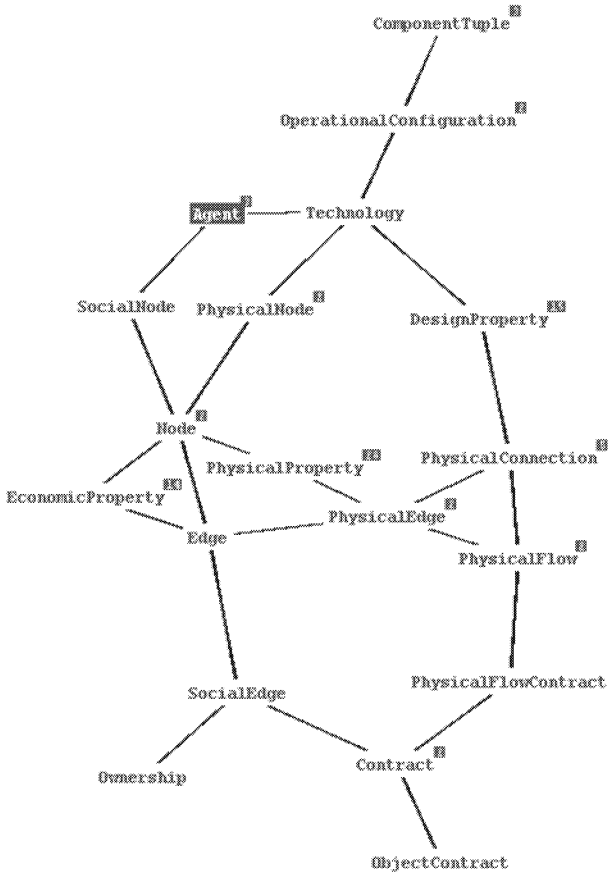


Figure 2. Fragment of the Ontology, Showing the Main Classes of the Socio-Technical System (Social Nodes and Physical Nodes) as well as the Links and Properties

have decision-making abilities and can interact with other agents, forming a social network. Examples are the electricity production companies, the world market, the power exchange, etc.

- **Technology:** A physical installation that transforms physical flows. Technologies cannot make decisions themselves. They operate in a certain operational configuration, which describes which flows go in and which flows go out. Technologies are connected with other technologies forming the physical network. Examples are a coal power plant, a wind farm, uranium delivery, etc.
- **Edge:** connections between nodes (given that both agents and technologies are nodes). Social edges (such as economic contracts) connect social nodes, while physical connections (such as cables or pipes) and physical flows (such as flows of electricity over time) can only occur between physical nodes.
- **Properties:** Both nodes and edges can have physical, economic and design properties. These properties, such as voltage, price, or maintenance costs, describe the elements that they are attached to.

The fragment of the ontology (Figure 2) shows the specific classes and properties used in this model. It is only a small part of the full ontology, with other classes such as data flow, transport contract, industrial plot, an extensive hierarchy of

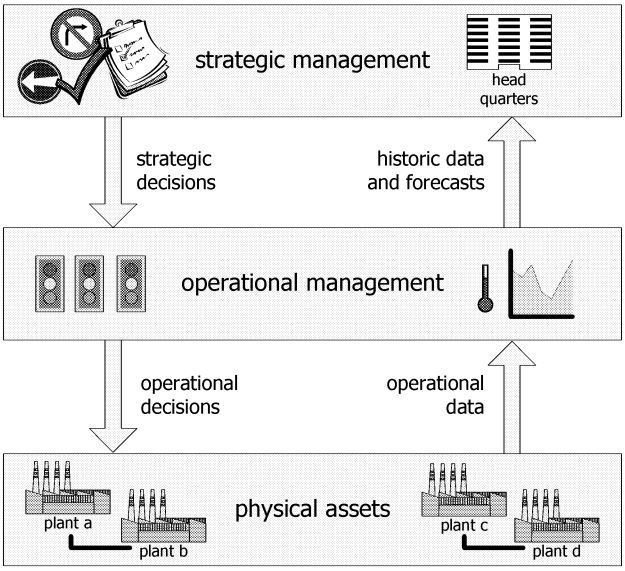


Figure 3. Three Layer Model (Chappin, 2006)

labels for goods, and more properties such as GIS locations, phase state, voltage, composition, distance and transport modality.

Besides this ontology structuring its concepts, a model is proposed to structure the socio-technical system itself. In the Three Layer Model, the social and technical subsystems are distinguished as well as their interaction. The Three Layer Model, shown in Figure 3, defines the interdependencies of actors and their physical assets. Actors in the real world are modeled as agents that manage on a strategic level and on an operational level. Agents thus act on these two levels.

On the strategic level agents make and plan for long term decisions that can affect their long term performance. Such decisions can be investment decisions, but also consideration of its objectives. On the level of operational management, daily procedures are followed and decisions are taken on a regular basis. This includes negotiation and contracting, buying resources and selling products, for example. Recall that actors and physical assets are in the ontology represented as social and physical nodes respectively, more specifically as agents and technologies.

Not only the distinction between agents and its physical assets are apparent in this model, but also the interaction that occurs between the three layers. The ontology concepts are merged with the three layer concept to formalize this interaction (see Figure 4). Strategic and operational management, as mentioned in the Three Layer Model, are parts of each agent. Both are distinct and have decision-criteria originating from the agents' *identity*, here defined as the characterizing set of properties of an agent. An agents' management style follows from these properties. Interaction is translated into different types of edges (depicted as the three arrows in Figure 4).

On the top layer, the decisions of agents can lead to changes in its identity, but also to changes in the ownership of physical installations. That means that **ownerships relations**

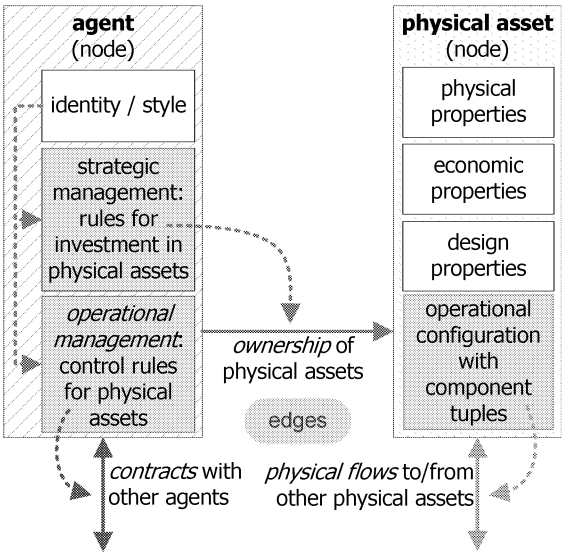


Figure 4. Interaction of Agents and Physical Assets: Extending the Ontology for Agents that Invest in and are in Control of Physical Assets

are defined at the strategic level. On the intermediate layer, agents negotiate and engage into agreements with other agents on the physical exchange of materials for a certain price and with certain conditions. This interaction is structured in economic **contracts**. Agents negotiate through unsigned contracts and when both agents have signed they have reached an agreement. On the lowest layer, interaction has a physical nature. The interaction of physical assets is by **physical exchange of materials**. Interaction on all those levels is in the ontology grasped in edges.

To summarize, social and physical nodes are distinguished. Social nodes are called agents; agents act by strategic management to change its identity and its ownership of physical nodes, also called technologies. Agents act by operational management to engage in contracts and physical installations constitute a physical network in which materials are exchanged. Actions are based on economic, physical and design properties of the agents and physical installations.

OPERATIONALISING DECISION-MAKING OF ELECTRICITY PRODUCERS

In this section, the decision-making of electricity producers is operationalised structured according to and based on the presented conceptual framework. As discussed in the previous section, agents act on two levels. This section focuses on the electricity producing agents, but some remarks are made on the other agents as well. An overview of the agents is drawn in the inner circle of Figure 5. Next to six electricity producers the government, consumer, other industry, power, fuel and emission markets and environment are modeled as agents as well. In this section, strategic management and operational management are separately discussed.

Strategic Management Of Electricity Producing Agents

Electricity producing agents' strategic management is the decisions to invest or disinvest in power plants and the criteria to do so. In reality this is a process in which preferred type and timing of investments and disinvestments are made together. For practical reasons, it is chosen to split this process in three consecutive parts. The following decision scheme is constructed:

1. deciding what power plants should be dismantled
   a. because the technical lifetime is passed
   b. or because it is a cause for continuous losses;
2. deciding if investing in a new power generation facility is needed
   a. because old power plants should be replaced
   b. or because there is need for portfolio expansion (e.g., to accommodate growing demands);
3. deciding which type of electricity generation plant is preferred, only after the decision was made to invest.

The electricity producing agents all use this (dis)investment decision structure. In fact, the criteria used to base decisions on are the same for all electricity producing agents. The only unique part is the relative importance given to the different criteria. Therefore, actual decisions can be different between the electricity producing agents. We will now focus on part three of the decision scheme: *deciding on which power plant type is preferred.*

It is assumed that electricity producing companies use more than one criterion to decide on their preferred power plant type. Therefore, a multi-criteria analysis is used to model the selection of the preferred power plant type. In general, in a multi-criteria analysis a number of alternatives are evaluated to more than one criterion. By weighing the different criteria, the best scoring alternative is chosen.

The set **t** contains k weight factors – actual values – that rank the importance of criteria of the multi-criteria analysis. The ranking in **t** is different for every electricity producer, making the agents unique in their decisions: they have a unique management style or preference. It is important to note that *only* **t** is agent-dependent. The other parts of the decision process are the same for all agents. The set of factors for the multi-criteria analysis is represented in (1):

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_k \end{bmatrix} \quad (1)$$

Five criteria are used, namely the yearly expected profitability, nuclear friendliness, market adoption, nuclear fear and the availability of locations. The agent-based approach makes it possible to include the softer criteria, such as nuclear fear. Electricity producing agents all evaluate those five criteria, thus k equals five, but they assign different levels of importance to these criteria. Those levels are grasped in **t**.

Let **a** be the alternatives that are evaluated in the analysis (the technologies that represent possible power plants with their economic, physical and design properties and possible operational configurations) and assume that m alternatives are evaluated then **a** can be represented as (2):

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \quad (2)$$

Five alternatives are evaluated (m therefore equals 5): a coal fired power plant, a natural gas power plant, a wind farm, a biomass power plant and a nuclear power plant. All electricity producers evaluate the same set of alternatives.

Next, a score matrix **S** is build in which the score of each alternative on all criteria is listed (see (3)).

$$\mathbf{S} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,k} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,k} \end{bmatrix} \quad (3)$$

Agents have the ability to calculate what the score is of all evaluated alternatives on the criteria. In other words: agents are able to calculate **S**. The ontology is used in the calculations: the data needed of the alternatives is represented in technology objects with physical, economic and design properties. For instance, the expected yearly profitability is calculated on basis of the economic properties of the different power plants, fuel consumption, fuel prices, etc. All data necessary is expressed in concepts defined in the ontology.

After filling **S** with scores, a normalization procedure is executed in order to make the criteria comparable. Per criterion, the 'best' score now equals 1 and the 'worst' score equals 0. So the most profitable alternative scores 1, the least profitable alternative scores 0. Now the scores are not related to chosen units anymore and the scores on different criteria are comparable. Matrix $\mathbf{S}^*$ contains the normalized values of matrix **S**, represented in(4).

$$\mathbf{S}^* = \begin{bmatrix} s^*_{1,1} & s^*_{1,2} & \cdots & s^*_{1,k} \\ s^*_{2,1} & s^*_{2,2} & \cdots & s^*_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ s^*_{m,1} & s^*_{m,2} & \cdots & s^*_{m,k} \end{bmatrix} \quad (4)$$

The normalization procedure is as follows. Let C be the subset of elements {1, 2, …, k} for which holds that the criteria t_C for all elements in C is positively oriented: a higher score s means a better score. For the elements {1, 2, …, k} not in C this is not the case: in other words, those criteria are negatively oriented. Given this subset c, normalization takes place by Equation (5). The normalized scores fill $\mathbf{S}^*$.

$$s^*_{i,j} = \begin{cases} \dfrac{s_{i,j} - \min\limits_{l\in\{1,2..m\}} s_{l,j}}{\max\limits_{l\in\{1,2,\ldots,m\}} s_{l,j} - \min\limits_{l\in\{1,2,\ldots,m\}} s_{l,j}} & j \in C \\ \dfrac{\max\limits_{l\in\{1,2..k\}} s_{l,j} - s_{l,j}}{\max\limits_{l\in\{1,2,\ldots,m\}} s_{l,j} - \min\limits_{l\in\{1,2,\ldots,m\}} s_{l,j}} & else \end{cases} \quad (5)$$

with $j \in 1,2...,k$

After normalization the scores are comparable: the 'best' scoring alternative has value 1, the 'worst' scoring alternative has value 0. Note that **S*** is the same for all agents, since they evaluate the same alternatives using the same criteria. However, the scores can now be weight by multiplication with **t**, the vector containing agent-specific preference values. After weighing, the alternative a_j corresponding to the highest score r_j is selected following Equations (6).

$$\mathbf{r} = \mathbf{S}^*\mathbf{t} = \begin{bmatrix} s^*_{1,1} & s^*_{1,2} & \cdots & s^*_{1,k} \\ s^*_{2,1} & s^*_{2,2} & \cdots & s^*_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ s^*_{m,1} & s^*_{m,2} & \cdots & s^*_{m,k} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_k \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} \quad (6)$$

$$r_j = \max_{i\in\{1,2,\ldots,m\}} r_i$$

Since r_j has the highest weight score, the chosen alternative is the j^{th} element of **a** (element a_j). Although **S*** not static over time, **S*** is equal for all agents at a certain point in time. Only the weight factors **t** differ, possibly leading to different selections for the preferred alternative power plant type, a_j. If for some weight factors, 0 is chosen, the criterion does not play any role in the decision. In this case, the criteria used are agent-specific as well. One could therefore, with the same structure also opt for an agent-specific set of criteria.

Operational Management Of Electricity Producing Agents

Besides strategic behavior, electricity producing agents also must exhibit operational behavior. As said in conceptual terms, by operational actions agents engage in negotiation and contracts. In general, a choice is made for demand-driven trade: negotiation always starts by the party that has demand for a good or service. Negotiation can lead to contracts and use the concepts defined in the ontology as language. For instance, if an agent needs coal, it will ask around for contracts for the delivery of the good called coal (all these concepts are defined in the ontology, recall the previous section).

Electricity producing agents perform the following consecutive steps:

1. bid on power exchange
2. acquire resources
3. acquire CO_2 emission rights

In these steps, the agents base their contracts on data of their physical assets: for coal power plants, for example, the required resource is coal. This is expressed in the operational configuration of this technology.

By interaction with the power exchange, contracts are setup to deliver electricity. More details can be found below. The amount of resources needed is based on its electricity production and contracts are setup to acquire the correct types and amounts of resources.

Electricity producing agents acquire CO_2 rights in contracts as well. This is different than the other resources, since it is no physical good. Since this process is demand driven as well, the government sells rights at no cost, based on an

allocation scheme. More rights can be bought on the emission rights market. Details on allocation, can be found in more detail in (Chappin and Dijkema, 2007b).

Operational Management Of Other Agents

Non-electricity producing agents, such as the consumer, are modeled as aggregate agents: one agent represents all consumers and one agent represents government. All agents look at their physical assets to acquire resources. In other words, they also do step 2. The consumer, for example, requires the resource 'electricity' for its technology 'consumer installation' (a representation of the electricity demand) and will look for agents that have the technology to supply them with this resource.

The electricity market agent accepts bids from electricity producing agents per installation. A set of bids contain for n installations a reference to the installation (i), a bidding price (b) and the capacity of electricity to be sold (c). The agents bid marginal cost and full capacity. The electricity market also requests import price, import capacity and aggregate demand.

By a market clearing model, the market algorithm selects how much each installation (*i*) will produce (*a*) and at what price (*p*) and records that in a set of contracts. The developed market clearing model has been described in detail in (Chappin and Dijkema, 2007b). Based on these contracts, which are social edges as described in the conceptual framework, agents transfer money and operate their power plants to deliver exactly the amount of electricity contracted.

All agents are demand driven and they negotiate with other agents based on observations of their physical installations. The input-side of the operational configuration of power plants gives information on the types and amounts of resources needed. Agents retrieve that information when needed.

Some modeling choices were made to make sure the physical representation is accurate. First, the environment is modeled as an agent and is also demand driven, which means that it contracts all emissions. It also is the supplier of air. Next, all primary resources come from one agent representing the world market. Those resources are essentially unlimited and prices are exogenously determined. Third, after negotiation, contracts are signed and agents are responsible for the operational actions on their physical assets, such that the contracted amounts are sent and received. Note that an electricity producer that does not need fuels or does not have emissions, consequently does not sign a contract with the world market or the environment, respectively. Finally, the agents themselves check that their physical flows of materials coincide with the settings in their physical installations (Nikolic *et al.*, in print) so a mass balance is ensured.

FROM PARAMETER SETS TO AGENT BEHAVIOR

In this section, a number of agents used in the electricity case is described (Chappin and Dijkema, 2007a; 2007b). An overview of the model's social and technical systems can be found in Figure 5. This picture is structured according to the Three Layers Model and the structure of interaction and (recall Figure 3 and Figure 4).

The inner circle contains the agents while physical installations are drawn in the outer circle. Ownership is shown as a line between an agent and its physical installation(s). Contracts between agents are drawn as lines between them. Physical exchange of materials is displayed between physical installations.

In the circle of agents, clockwise from three o'clock, there are one industry agent, three markets (electricity, CO_2 permits and fuels), the government, the environment, the consumer and finally six electricity producers. For the physical system Figure 5 shows, again clockwise from three o'clock, one industry installation, five fuel delivery installations, one installation for the environment, one consumer installation, and thirteen power generation units of different types.

The model currently contains a static number of agents. The establishment of contracts depends both on the physical layer (e.g. does a facility need natural gas to produce electricity?) and the style and preferences of a particular agent (will I invest in natural gas fired power plant, or is a wind farm with lower return on investment also an option). Over time, changes in the contract layer do occur and as a result the social network is adjusted.

Based on the contracts arranged via the power exchange market or by bilateral agreement, agents operate their power plants to supply electricity to the consumer and acquire resources from world markets. Their discrete and autonomous investment or disinvestment decisions lead to changes in the physical system. Long-term, the power generation portfolio, overall emission levels of power

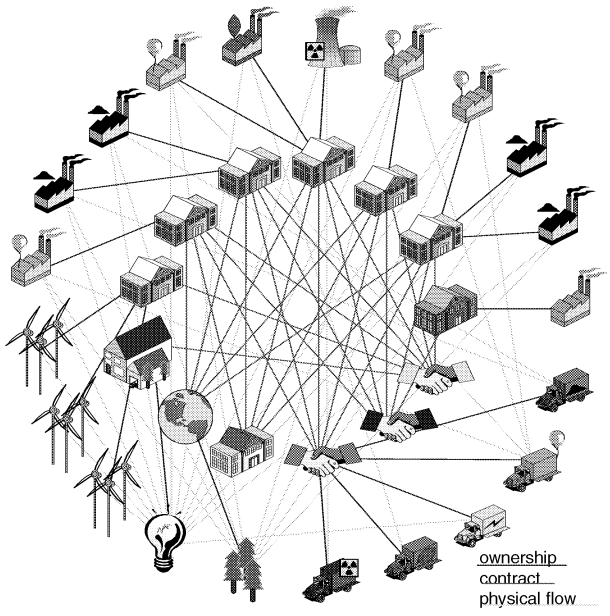


Figure 5. Simulation of the Socio-Technical Network. The Inner Circle Consists of the Agents and the Outer Circle of the Physical Installations

production, electricity prices and emission credit prices are also affected.

On a disaggregate level, agents decide based on their own criteria. Agents evolve based on their actions since their decisions have long-long term effects; many factors are used in decision criteria, reflecting the present and past. Because vector **t** (the set of factor values for decisions) is unique for each electricity producer, agents evolve over unique patterns, shaped by external factors.

The parameter sets **t** of electricity producing agents are defined and balanced as such that they represent typical companies as observed in reality. Each set defines the management style of a company, which is part of its identity.

Three examples are now discussed. Electricity producer 1 is the *'risk aversive market-player'*. This agent tries to spread risks by investing in different technologies and withholding investment until a good business case is certain. Electricity producer 2 is a typical *'green energy company'*. For this company, its environmental mindedness is seen as key for consumer attraction. Cost-effectiveness is less important, because of an expected volume effect of the positive, green image of the company. Since this company expects that CO_2-emissions are the long term environmental indicator, nuclear power is considered more seriously than by the other companies. Grants are exploited where possible to make a business case. Electricity producer 3 is the *'rational, profit-oriented, big player'*. This agent is careful to the fact that it will rely on known facts: power plants that have been profitable for the last decades will be sufficient in the future as well.

These three market players are characterized by their parameter set **t** and, as a consequence, the agents differ as well in their simulated behavior. That is observed from the most important emerging property: the power plant portfolio of electricity producers. The average developments of the portfolio of three electricity producers have been drawn in Figure 6. These are averages over more than 900 runs in which external developments differ.

Because of differences in the investment and disinvestment decisions, over time the (relative) amounts of power plant types (forming the portfolio) develop in a way that is characteristic for the agent. It shows, for example, that the portfolio of electricity producer 2 (the green energy company) is much more sustainable than that of electricity producer 1 (the risk-aversive market player). Electricity producer 3 (the rational, profit-oriented, big player) is a cause for large emissions by using coal and natural gas as main energy sources.

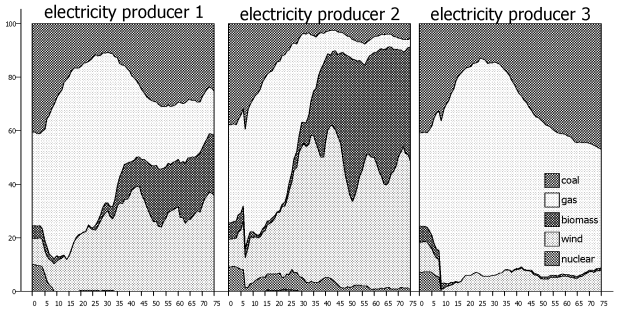


Figure 6. Evolution of Individual Agents' Portfolio

Although all agents live in the same world and thus are impacted by the same external factors, their actions lead to very different power generation portfolios, because of different styles. These developments are found to be interdependent: the portfolio development of electricity producer 1 depends on the other electricity producers. As a consequence it cannot be concluded that if there existed a world with only 'green energy companies' alike that power generation would be as sustainable as shown here. The decisions are disaggregately made, but not independent of other market players. However, since on average the differences are significant (and that is also the case in reality), it is valuable to model this disaggegrately.

Strikingly, it was found that in absolute sense the portfolio of electricity producer 3 was largest – he had the largest total capacity – and therefore had the most effect on the emerging total portfolio. The result was that, although electricity producer deviated most from the rest, it was a key determiner of the total portfolio, which was found quite unsustainable. More model results and conclusions on the case can be found in (Chappin and Dijkema, 2007a; 2007b).

The parameter sets **t** that weigh the criteria of power producers is part of the uncertainty in the model, since the set of investment motives selected by power producers is unknown. To deal with this fact, a wide variety of power producers' parameter sets was selected. However, since they are interdependent, as mentioned earlier, also combinations of style sets will be explored with scenario analysis in the future.

CONCLUSIONS AND FINAL REMARKS

A conceptual framework, consisting of an ontology for socio-technical systems and the interaction within it has been presented and the decision-making process was operationalised for electricity producers. Electricity producers are modeled as agents that own and operate power plants which are their physical assets in the physical layer. The system is modeled by describing the individual decision-making of the agents so that the system behavior emerges from this.

The agent-based approach is powerful because it makes it possible to model social interaction of agents as well as an adequate representation of the physical system. The model can easily be extended or adjusted, for example to the impact of policies or to include new types of power plants.

The ontology is helpful to formalize the concepts used in the decision-making process of the electricity producing agents. A similar approach can and is followed in other domains and the use of a shared ontology allows re-use of building blocks and sharing source code of the simulation models.

REFERENCES

Aldea, A., R. Banares-Alcantara, L. Jimenez, A. Moreno, J. Martinez and D. Riano. 2004. "The scope of application of multi-agent systems in the process industry: three case studies." *Expert Systems with ApplicationsExpert Systems with Applications* 26(1): 39-47.

Axelrod, R. 1997. Advancing the Art of Simulation in the Social Sciences. *Simulating social phenomena*. R. Conte, R. Hegselmann and P. Terna. Berlin ; New York, Springer: 21-40.

Borshchev, A. and A. Filippov. 2006. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools.

Bussmann, S., N. R. Jennings and M. Wooldridge. 1998. *Multiagent Systems for Manufacturing Control*. Berlin, Springer.

Chappin, E. J. L. 2006. *Carbon Dioxide Emission Trade Impact on Power Generation Portfolio, Agent-based Modelling to Elucidate Influences of Emission Trading on Investments in Dutch Electricity Generation*. Delft, Delft University of Technology.

Chappin, E. J. L. and G. P. J. Dijkema. 2007a. An Agent Based Model of the System of Electricity Production Systems: Exploring the Impact of CO_2 Emission-Trading. *IEEE SoSE: Systems of Systems Engineering*, San Antonio, Texas, USA, IEEE.

Chappin, E. J. L. and G. P. J. Dijkema. 2007b. On the Impact of CO_2 Emission-Trading on Power Generation Emissions. *10th International Conference on Technology Policy and Innovation*, Stavanger, Norway, ICTPI.

Dam, K. H. v. and Z. Lukszo. 2006. Modelling Energy and Transport Infrastructures as a Multi-Agent System using a Generic Ontology. *IEEE International Conference on Systems, Man, and Cybernetics*, The Grand Hotel, Taipei, Taiwan.

Dam, K. H. v., Z. Lukszo, R. Srinivasan and I. A. Karimi. 2007. Opportunities for Agent-Based Models in Computer Aided Process Engineering. *European Symposium on Computer Aided Process Engineering ESCAPE 17*, Bucharest, Romania.

Dam, K. H. v., I. Nikolic, Z. Lukszo and G. P. J. Dijkema. 2006. Towards a Generic Approach for Analyzing the Efficiency of Complex Networks. *IEEE International Conference On Networking, Sensing and Control*, Ft. Lauderdale, Florida, USA.

Epstein, J. M. and R. Axtell. 1996. *Growing artificial societies: social science from the bottom up*. Washington, D.C., Brookings Institution Press; MIT Press.

Jennings, N. R. 2000. "On agent-based software engineering." *Artificial Intelligence* 117(2): 277-296.

Laurikka, H. and T. Koljonen. 2006. "Emissions trading and investment decisions in the power sector - A case study in Finland." *Energy Policy* 34(9): 1063-1074.

Nikolic, I., G. P. J. Dijkema and K. H. v. Dam. in print. Understanding and Shaping the Evolution of Sustainable Large-Scale Socio-Technical Systems, Towards a Framework for Action Oriented Industrial Ecology. *Dynamics of Industrial Ecosystems*, Reviewed and accepted.

Nikolic, I., G. P. J. Dijkema, K. H. v. Dam and Z. Lukszo. 2006. General Methodology for Action-Oriented Industrial Ecology. *IEEE International Conference On Networking, Sensing and Control*, Ft. Lauderdale, Florida, USA.

Nikolic, I., G. P. J. Dijkema and M. A. Reuter. 2004. Proposed framework for the understanding of the Co-evolution of Industry and infrastructure. *Gordon Research Conference on Industrial Ecology*, Queens College, Oxford, United Kingdom.

North, M. J., N. T. Collier and J. R. Vos. 2006. Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit *ACM Transactions on Modeling and Computer Simulation*, New Your, New York, USA.

North, M. J. and C. M. Macal. 2007. *Managing Business Complexity - Discovering Strategic Solutions with Agent-Based Modeling and Simulation*. Oxford, Oxford University Press.

Olsina, F., F. Garces and H. J. Haubrich. 2006. "Modeling long-term dynamics of electricity markets." *Energy Policy* 34(12): 1411-1433.

Ottens, M., M. Franssen, P. Kroes and I. Van De Poel. 2006. "Modelling infrastructures as socio-technical systems." *International Journal of Critical Infrastructures* 2(2-3): 133-145.

Schieritz, N. and P. M. Milling. 2003. Modeling the Forest or Modeling the Trees - A Comparison of System Dynamics and Agent-Based Simulation. *21st International Conference of the System Dynamics Society*, New York.

Schleich, J., K. M. Ehrhart, C. Hoppe and S. Seifert. 2006. "Banning banking in EU emissions trading?" *Energy Policy* 34(1): 112-120.

Smith, A. 1776. *An Inquiry into the Nature and Causes of the Wealth of Nations*

Svendsen, G. T. 1999. "The idea of global CO_2 trade." *European Environment* 9(6): 232-237.

Svendsen, G. T. and M. Vesterdal. 2003. "How to design greenhouse gas trading in the EU?" *Energy Policy* 31(14): 1531-1539.

Vesterdal, M. and G. T. Svendsen. 2004. "How should greenhouse gas permits be allocated in the EU?" *Energy Policy* 32(8): 961-968.

Weiss, G. 2000. *Multiagent systems, a modern approach to distributed artificial intelligence*. Cambridge, Mass., MIT Press.

Wooldridge, M. J. and N. R. Jennings. 1995. "Intelligent agents: theory and practice." *Knowledge Engeneering Review* 10(2): 115-152.

AUTHOR BIOGRAPHIES

EMILE CHAPPIN holds a Bachelors and Masters Degree in Systems Engineering, Policy Analysis and Management – Delft University of Technology. In 2007 he started his PhD research on modeling and transition management of energy infrastructures at Delft University of Technology. E.J.L.Chappin@TUDelft.NL; http://www.chappin.com/en.

GERARD DIJKEMA is Associate Professor, Energy and Industry. A chemical engineer by profession – University of Twente – he holds a PhD from Delft University. His research focus is innovation in industrial networks. G.P.J.Dijkema@TUDelft.NL.

KOEN VAN DAM is a PhD researcher at the Energy and Industry group of Delft University of Technology. He received a Master's degree in Artificial Intelligence from the Vrije Universiteit in Amsterdam. Current research interests include agent-based modeling, generic ontologies, intelligent infrastructures and process systems engineering. K.H.VanDam@TUDelft.NL; http://www.koenvandam.com.

ZOFIA LUKSZO is Associate Professor in the Section Energy and Industry at the Faculty of Technology, Policy and Management of the Delft University of Technology. Her research interests include plant management and quality control. Z.Lukszo@TUDelft.NL.

VILLON – AGENT BASED GENERIC SIMULATION MODEL OF TRANSPORTATION LOGISTIC TERMINALS

Norbert Adamko
Valent Klima
Faculty of Management Science and Informatics
University of Zilina
Slovak Republic
Norbert.Adamko@fri.uniza.sk,
Valent.Klima@fri.uniza.sk

Antonín Kavička
Jan Perner Transport Faculty
University of Pardubice
Czech Republic
Antonin.Kavicka@upce.cz

KEYWORDS

Agent based simulation, generic model, transportation terminals, logistic systems, microscopic simulation.

ABSTRACT

The paper deals with the modelling of transportation logistic terminals operation (e.g. marshalling yards, factories, train care centres, airports, etc.) utilizing generic simulation tool Villon, which was developed using agent based simulation architecture ABAsim. Villon is complete simulation tool providing support for creation, configuration, experimentation and results evaluation tasks connected with simulation modelling of transportation terminals. Villon's unique features like its flexibility, detailed microscopic modelling of operation of various types of terminals, flowchart driven definition of operational procedures, user interaction, selection of decision strategies, 3D animation output as well as its agent structure and model creation process are explained. Project references and problems that can be addressed using presented simulation tool are also mentioned.

INTRODUCTION

The most important and evidently the most costly part of a logistic chain is transportation logistic. The transportation logistic itself is composed of a movement along the transportation route and of necessary manipulations (service processes) applied to transportation means and to transported commodities at specialised locations called transportation logistic terminals (e.g. marshalling yards, factories, train care centres, airports, etc.). Terminals belong to the most complex service systems involving sophisticated technological processes and are equipped with quite complicated and costly technical devices.

Simulation methods currently represent widespread techniques supporting optimisations and planning related to transportation logistic terminals. Presented simulation tool *Villon* supports tactical (middle-term) and strategic (long-term) planning related usually to infrastructural or operational proposals, which are supposed to guarantee optimal (or at least effective) behaviour of modelled terminal. Villon is a generic simulation model which supports microscopic modelling of various types of transportation logistic terminals containing railway and road infrastructures (e.g. marshalling yards, railway passenger stations, factories, train care centres, depots, airports, etc.). Using Villon, users are able to create detailed simulation models of terminal operation, define simulation scenarios, make experiments with the model and evaluate results of simulation runs in one integrated user-friendly environment.

The development of Villon was based on proprietary agent based simulation architecture ABAsim.

ABASIM SIMULATION ARCHITECTURE

The ABAsim (Agent Based Architecture of simulation models) provides means for design of flexible simulation models of complex service systems. In the following text we will try to explain only the basic properties of the ABAsim architecture, more in-depth explanation can be found in (Adamko et al. 2004).

In the ABAsim architecture, simulation models are composed of cooperating agents, which are organised in a hierarchical structure, which is a typical property of transportation systems and many other service systems.

The simulation models of simple systems could be composed of only one agent; however the simulations of complex service systems are obviously connected with multi-agent approach using the agents within some organizational structures. One of the goals of the architecture is to try to model not only the behaviour of the modelled system but also its internal structure as close as possible.

Each agent is composed of internal components responsible for sensorial, executive and communication activities of an agent. Each agent contains one *manager* (responsible for control, decision making and communication tasks, its behaviour can be described by Petri nets) and a set of *assistants*. Based on the roles of assistants, we can divide the assistants into three groups, each group containing one instant and one continual assistant: *solvers* (advisor, scheduler), *effectors/actuators* (action, process) and *sensors* (query, monitor). Continual assistant execution is invoked strictly only by agent's

manager.

All communication, internal inside-agent (between manager and other components) and also external inter-agent, is realised exclusively by messages.

Modelled system is in ABAsim architecture represented by permanent hierarchical structure of agents, which are responsible for their respective tasks. The agent structure does not change during execution of the model; no new agents are created nor are any agents released from the structure during execution of the model. Single entities are not represented by agents (as many other architectures do), but agents manage many entities, which share the same behaviour. In the original ABAsim architecture, model entities (customers, resources) are considered to be unintelligent, without any initiative and are only manipulated by agents' effectors.

VILLON SIMULATION TOOL

The properties of ABAsim architecture enabled to design and implement complex generic simulation tool for modelling of transportation terminals called Villon. The simulation tool Villon allows users (professionals in the field of logistics) to create simulation models of transportation terminals, to run prepared scenarios as well as to evaluate results of simulation runs, all without the need to write a single line of program code – utilizing only Villon's user-friendly interface. The creation of complex simulation models of logistic systems, of course, requires a certain level of experience and knowledge. However, using the Villon simulation tool, even less experienced user is able to create simulation models of simple logistic systems within a short period of time (few days).

Even though the development of this tool was motivated by the ambition to create complex simulation model of a railway marshalling yard, nowadays Villon is able, thanks also to the architecture flexibility and its other valuable properties, to support modelling of various types of logistic transportation terminals (e.g. railway passenger stations, train depots, factories including road transportation, ground handling within airports etc.).

Villon is a complete generic simulation system; it provides the user with comfortable user-friendly editors to edit all needed data to run a simulation model, supports customisation of many aspects of simulation runs, offers animated output of modelled activities in 2D or 3D view as well as extensive set of post-run evaluation tools (including statistics, graphical protocols and others).

In comparison with other commercially available simulation tools used for modelling of railway operation (among many, we can mention RASIM (HaCon 2007), RailSys (Bendfeldt et al. 2000) or OpenTrack (Nash & Hürlimann 2004)), Villon offers its users more precise modelling and visualisation of infrastructure (no schematic approach), detailed modelling of resource (personnel, engines) activities, as well as the possibility to model railway and road traffic in single model to examine mutual interference. On the other hand, due to its focus on modelling of operation of terminals (nodes), Villon lacks support for modelling of large-scale railway networks or timetable creation abilities.

In the following text some features of Villon simulation tool and the process of simulation model creation will be presented.

MODEL AGENT STRUCTURE

Villon is a multi-agent simulation system composed of more than a dozen of agents responsible for various tasks, organised in a hierarchical manner (Fig. 1). *Dispatcher* represents a topmost agent responsible for all model agents, which is directly superior to *Surrounding* agent (reflecting relevant processes within surroundings of examined system, e.g. arrival of customers), *Service* agent (responsible for the execution of user-defined technological procedures) and *User Cooperation* agent, which stands for humans interacting with the model during run-time (human experimenter can easily take responsibility for chosen decision making tasks, e.g. choosing which track should be assigned to incoming train). Common *Resource* agent governs a set of agents (*Infrastructure*, *Personnel* and *Engines*) responsible for allocation, work and release of respective resource types. Group of *Operation* agents is responsible for modelling of operations related to various types of modelled systems – e.g. movement of vehicles using rail or road infrastructures (in competence of *Transfer* agents – if required Villon provide the possibility of precise modelling of train movement dynamics, considering vehicle type, track properties, slope, curves, air resistance and many other factors utilising continuous simulation techniques supported by ABAsim architecture), sorting in marshalling yards (under supervision of *Sorting* agent), group operations at factory sidings, road crossing traffic and others.

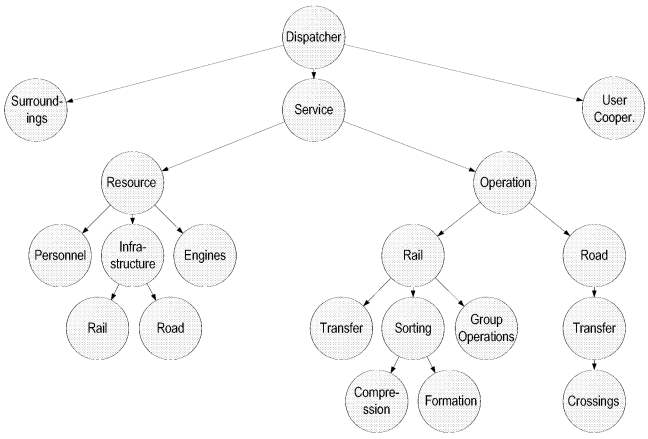


Figure 1: Hierarchical agent structure of Villon

Utilization of ABAsim architecture in the development of the tool gave programmers the possibility of comfortable modelling functionality augmentation of the tool (e.g. by adding more agents and sub-models to the existing model structure) without the need for extensive program code changes in existing parts of the model. For example, programmers can simply add the support for another type

of service resources (e.g. fork-lifters) by adding the responsible agent to the hierarchical structure. Even at present Villon is being continuously improved by adding more modules (agents and sub-models) and by modification of existing agents and their assistants. During the configuration of a simulation model, experimenter is able to specify modules that the model will use, which leads to corresponding structure of agents (agents responsible for not requested functionality are not included in the model structure).

CREATION OF SIMULATION MODEL

The creation process of simulation model of complex logistic system cannot be successfully accomplished without proper data related to various aspects of modelled system. Following the internal structure of the simulation system, we can divide needed data into three main categories – resources, customers (orders) and operation (services executed).

Resources

Since resources represent vital element in operation of any logistic terminal, special attention has to be paid to their modelling. The infrastructure model is created by direct transformation of a map (plan), which is available either on paper, or in electronic form – an accurate, not only simplified (schematic) model of infrastructure is used in the simulation and therefore we can evade distortions caused by rough approximations. Mobile resources (e.g. personnel, engines) are modelled individually, respecting their working hours, profession and other properties.

Management of resources, which is under supervision of specialised resource agents, can be modified thanks to the ABAsim architecture properties by the users in many ways, for example the users have the chance to choose suitable algorithm for personnel assignment. This is in Villon implementation translated as a substitution of advisor component of Personnel agent for another one (which implements alternative algorithm of personnel assignment) from the pre-programmed palette of available advisors. The same principle is applied also to other Villon agents. If there is a need to alter agent's behaviour in greater extent, the user has the possibility to modify the Petri net describing agent's behaviour; however this task requires basic knowledge of Petri nets and certain algorithmic skills (therefore it is mostly applied by programmers).

Customers

Data about customers (e.g. trains, cars) are provided in the form of arrival timetables with the possibility of probabilistic modification of their properties (arrival time, loading, composition, ordering of groups, etc.). Timetables could be imported from XLS files (many terminal disposition systems are able to export to this format), which greatly reduces time required to input the data.

Operation

The main operational procedure inside logistic terminals (similarly as in any other service systems) is serving of customers entering the system. Since Villon is generic model and has to support modelling of various logistic terminal types, it does not contain any hard-coded service procedures – simply said, Villon itself is not able to perform any task without proper "program", which is entered in a form of a flowchart by the user during creation of the model (Fig. 2). Flowcharts are composed of activities, which represents single tasks executed during serving of customers (e.g. loading, resource assignment, brake testing, etc.).

Villon contains a set of predefined, so called *template activities*, which are prepared by programmers and cover the whole spectrum of functions found in operation of transportation terminals (there are around 30 template activities). Template activities are not to be used directly, they serve only as a template for creation of *derived activities*. Template activities define parameters and mandatory resources for each type of derived activity. User has the chance to create nearly unlimited amount of derived activities and customize them by modification of their parameters and resources. Each derived activity contains:

- *Set of resources*, required for the activity to be executed. For example activity *'Simple movement'* defines resources *Infrastructure type* (e.g. electrified tracks) and *Engine type* (e.g. shunting engine). Besides mandatory resources defined in template activity, user can specify additional resources. Each resource is identified either by its profession (any resource with given profession, which represents abilities of the service resource, can be used) or by its name (only the specified resource can be used).
- *Set of parameters*, which more closely specify its execution. In previously mentioned activity *'Simple movement'* we have to, among other, define parameters like *'move speed'* or *'priority'*. By *'Technical inspection'* activity completely different set of parameters has to be defined – e.g. *'duration per wheel'* or *'duration per car'*.

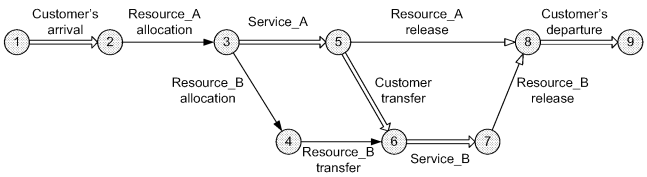


Figure 2: Example of a flowchart defining technological procedure of serving a customer

Parameterized derived activities are composed to flowcharts, which define succession and mutual dependence of activities in a service process. Defined derived activities are reusable and can be used in more than one flowchart. Flowcharts are created in a comfortable graphical editor with support for automatic validation of entered flowcharts (guarding required succession of some

activities and appropriate resource handling). Ready flowchart is then assigned to a customer (e.g. a train or truck). Once defined flowchart can be reused – the same flowchart can be applied to different customers with the same attendance procedure.

During simulation run, specialised *Service* agent interprets defined flowcharts – each activity is simply translated to a message that is at a proper time (respecting the order and dependences between activities defined in the flowchart) sent to the agent, which carries out the activity, reflecting defined resources and parameters.

Simulation model data structure

In Villon each model consists of *Model Data*, *Run Properties* and scenarios (called *Configurations*).

Currently Villon *Model Data* contains around 25 physically distinct types of data that completely describe modelled system operation and are organised in hierarchical manner based on their mutual dependencies (starting with infrastructure, followed by other resource types, service activities, service procedures and at last customers). Model data are loaded respecting the hierarchy – high level data are loaded first, which enables to perform pre-run validation of entered data and also allows comfortable editing. For example Villon checks if incoming track for a train, which is defined at lower level (when defining timetable) is present in infrastructure data and on the other hand, Villon's timetable editor allows user to select only such an incoming track that is defined in infrastructure data as a track suitable for arrival of trains.

Run Properties portion of model define attributes of simulation run – duration, animation settings, simulation protocol options and cooperation options.

User has the chance to create many variants (represented by distinct data files) of every data type. Simulation scenario is then simply created by selection of single variant of every data type (e.g. the user selects one variant of infrastructure, one variant of personnel and so on (Fig. 3)). Scenario created such a way is associated with a specific name and can be stored in the database of scenarios (configurations).

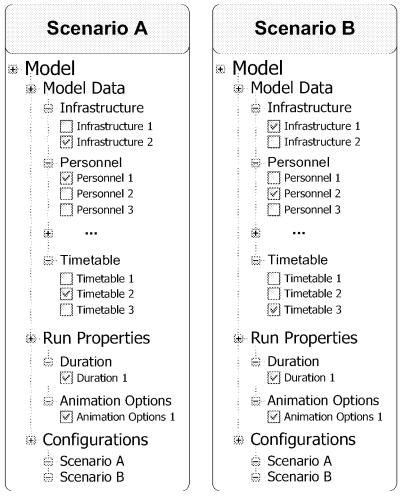


Figure 3: Villon model data structure and scenarios

SIMULATION RUN

Once all needed data are collected and the scenario is chosen, the simulation run can be started. During the simulation run animation of all movements of vehicles (train sets, cars, airplanes, locomotives, etc.) is presented to the user. User can choose between two or three dimensional view of the scene (Fig. 4).

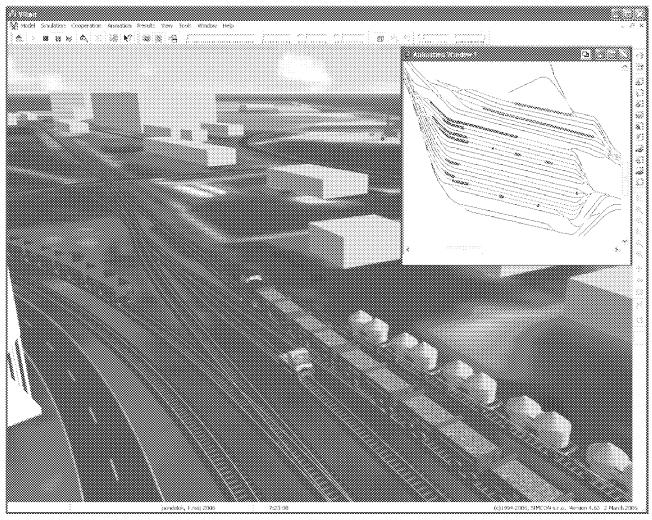


Figure 4: Example of 2D and 3D animation in Villon

RESULTS EVALUATION

As the simulation proceed user can follow a "live" development (change of values) of selected characteristics in graphical presentation (e.g. status of personnel or utilization rate of selected locomotive). Another type of output information is the overall information retrievable after termination of the simulation run. To this purpose, a detailed protocol on simulation is generated during the simulation run. These protocols can be processed separately and a sought for information can be retrieved from them. Various statistics are obtainable from the palette of pre-defined evaluations. The palette is open for adding new items according to the wish of the user. Besides graphical presentation of simulation results using time dependent reports on utilization of resources, waiting times, etc. (Fig. 5), statistical evaluations are also provided, in the form of tables, graphs and charts (including PERT analysis of flowcharts describing operational procedures, resource utilisation statistics and many others).

Villon also offers the chance to export all collected information to the XLS file for further processing using spreadsheet editor.

Important feature of Villon is the possibility of deploying a special Viewer version executable of the simulation tool to the customer. This Viewer version enables customers (e.g. the owner of examined terminal) to execute simulation runs prepared by model designers and also exploit complete range of run-time and post-run evaluation options Villon offers. Of course, Viewer does not allow the customer to change any model data, which could have unpredictable consequences.

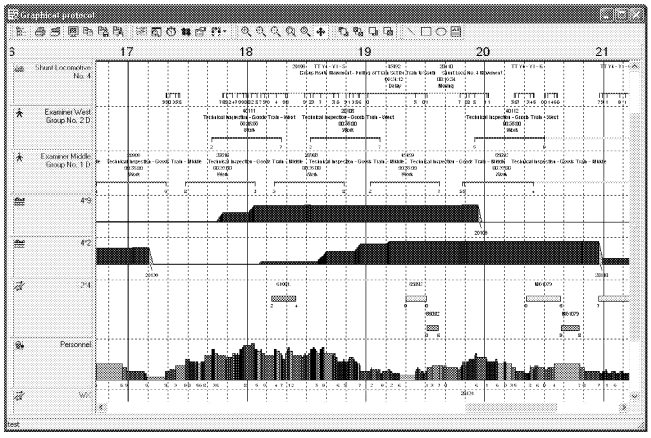


Figure 5: Graphical time dependent reports example

APPLICATIONS OF VILLON

The tool Villon was applied in commercial environment within simulation studies of marshalling yards in Austria (Vienna, Linz), in Germany (Hamburg Alte Süderelbe, Oberhausen-Osterfeld), in Switzerland (Lausanne, Basel), and in China (Mudanjiang, Harbin). Other applications of the discussed tool concentrated for example on investigation of new proposed design of railway depot in Ulm (Germany); verification of operation of the Prague Masaryk passenger station (Czech Republic), or proofing of the internal railway-road traffic concepts inside the car production company Volkswagen Bratislava (Slovakia). Among other applications in industry (mostly connected with production increase in modelled facilities) following can be mentioned – the factory sidings of the chemical plant BASF Ludwigshafen (Germany), the internal railway traffic of the paper producing company SCA Laakirchen (Austria) and steel production companies Voest Alpine Linz (Austria) and Teesside Cast Products Middlesborough (United Kingdom).

The detailed description of one relevant simulation study is available in (Kavička et al. 1999).

CONCLUSIONS AND FUTURE WORK

The substantial decisions in the field of transport, respecting the contemporary level of information technologies and simulation methodologies, should not be adopted without the modelling of their consequences. The software tool Villon supports microscopic simulations of the most complex and large-scale transportation terminals and enables not only to investigate the consequences of adopted decisions but also (by means of the reasonable sequence of experiments) to choose the best solution and to save the financial resources.

The generic character of simulation tool Villon brings one important consequence: once built, verified and validated model of one specific node can be used repeatedly in seeking for solutions of a different type. For example, the same simulation model can be utilized for investigation related to optimal reconstruction of a terminal (long-term planning); on the other hand it can be also useful during the real reconstruction in order to study operation within different reconstruction stages (medium-term planning).

Simulation tool Villon was originally determined for visual dynamic micro-simulations of railway marshalling yards; however its flexible hierarchical software architecture enabled to extend its scope to other kinds of transportation terminals. The tool was applied to different kinds of terminals always with reasonable economical profit.

Villon is in development for more than 10 years. Since the beginning Borland Delphi rapid application development tool is used as an implementation platform (utilising Object Pascal programming language).

Development of Villon is an ongoing process, current efforts are aimed at the improvement of the road simulation module utilising new features of ABAsim architecture – actors, which will be used for modelling of intelligent entities (e.g. cars, trucks). Another scope of our interest is the integration of storage modelling modules (agents responsible for modelling of storage operations) into Villon design.

ACKNOWLEDGEMENT

This work has been supported by the Slovak Grant Foundation under grant No. 1/4057/07 "Agent oriented models of service systems" and by the National Research Program of Czech Republic under project No. MSM 0021627505 "Theory of Transportation Systems".

REFERENCES

Adamko, N., Klima, V., Kavička, A., Lekýr, M. 2004. "Flexible hierarchical architecture of simulation models". In *Proceedings of European simulation and modelling conference*, Eurosis, Paris, 2004, pp.30-34, ISBN 90-77381-14-7.

Bendfeldt, J-P., Mohr, U., Müller, L. 2000. "RailSys, a system to plan future railway needs", *Computers in Railways VII*, WIT Press, 2000, ISBN 1-85312-826-0

HaCon. 2007. "RASIM - Simulation system for the analysis and planning of railway systems", *http://www.hacon.de/rasim_e/index.shtml*

Kavička, A., Klima, V., Niederkofler, A., Zaťko, M. 1999. "Simulation model of marshalling yard Linz Vbf (Austria)", In: *Proceedings of The international workshop on Harbour, Maritime & Logistics Modelling and Simulation*, SCS, 1999, pp. 317-320, ISBN 1-56555-175-3

Nash, A., Hürlimann, D. 2004. "Railroad Simulation using OpenTrack", *Computers in Railways IX*, pp. 45-54, WIT Press, 2004, ISBN 1-85312-715-9

INTELLIGENT AGENT MODELLING

Towards a psychological perception in agent modelling

Jean-Michel AUBERLET
INRETS-MSIS
2 Avenue du Général Malleret-Joinville 94114 Arcueil Cedex France
e-mail: jean-michel.auberlet@inrets.fr

INTRODUCTION

In a dynamic environment, the displacement of an agent corresponds to achieving a displacement from an origin to a destination by avoiding fixed and/or moving obstacles by taking account a set of constraints. The obstacles could be of the same nature that the agent or not. For example, in urban environment, a pedestrian moves along his path by avoiding the other pedestrians, cars, motorcycles... Furthermore the pedestrian abilities are limited such that the walk speed for example.

A first solution to the displacement of an agent is to determine (to plan) the shortest distance between the origin and the destination. But the planning of the path which corresponds to an optimal trajectory should be calculated by taking into account the motions of other agents (mobiles obstacles). Indeed, what an agent can do, the other agents can do too. Thus, due to the movements of the other agents, the agent could deviates from his planned trajectory. At this stage of this brief discussion the major stake arises. How do we take into account the movements of the other agents ?

A first approach to answer to this question consists to focus on the perceptive abilities of the agent. Indeed, according to the types of information that the agent perceives/receives, its representation of the environment will be different. In Figure 1, the agent only knows the position of the moving obstacle (model A) and he has sufficient information about the obstacle to predict its motion (model B). Thus the obtained trajectory of the agent will be different depending on the received information.

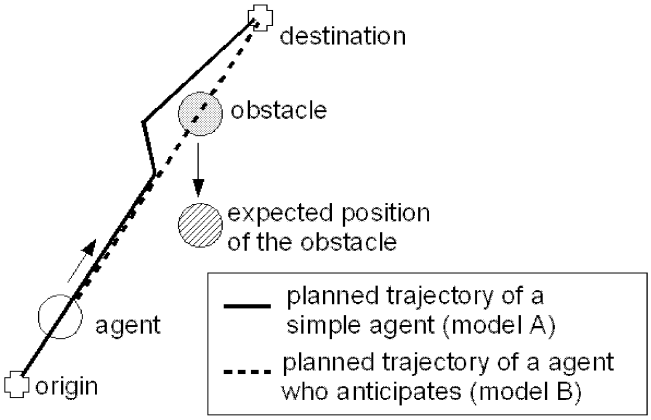


Fig. 1: trajectories according to the way the environment is perceived

These considerations illustrate the impact of the information (location, speed, direction...) that can perceive/receive the agent on the decision it takes to make its displacement. In this contribution, we do not focus on the way which the agent perceives its environment as through vision sensors for example. We only focus on what the agent could do with what he perceives. Thus this contribution aims to discuss about an old paper written by the psychologist Gibson in 1938 [Gibson and Crooks 1938]; to highlight how Gibson's concepts could be used in the problematic of the displacement of agents in a dynamical environment and to propose an agent model able to integrate the concepts developed by Gibson.

IF I CAN MOVE, THE OTHERS TOO

Under this funny title, there is major matter: what an agent can do, the other perceived agents (obstacles) can do too. Thus, it is important to take into account the possible actions of the others and these actions can strongly influence those of the current agent. In Gibson's paper, two points are worth highlighting in the proposed context.

Before this, it is useful to define the concept of locomotion in accordance with Gibson:
"*the primitive function of locomotion is to move the individual from one point of space to another, the destination (...) In the course of locomotion obstacles are met with, i.e. are seen, and the locomotion itself must be appropriately modified in order to avoid them ... Locomotion is therefore guided chiefly by vision, and the guidance is given in terms of path within the visual field of the individual such that obstacles are avoided and the destination ultimately reached. These concepts of destination, obstacle, collision and path should be applicable to any type of locomotion.*"

Based on this affirmation, Gibson defined the concept of "field of safe travel" which represents the set of possible paths that an agent can travel on. It is the factors which limit the field of safe travel that interest us. Among this factors Gibson identified moving obstacles and potential obstacles. For Gibson even the current location of moving obstacle may belong to the field of safe travel, it influences the field of safe travel in the following way: the obstacle influences the agent from "*the point where the obstacle will be when the agent comes closest to it, in other words from the point of potential collision*". Similarly, even the

potential obstacles do not belong to the field of safe travel, they can influence the agent since the point of potential collision could belong to the field, and thus the field should be modified.

Gibson's works are very interesting especially in his way to analyse the interactions between an agent and its environment. A first use of Gibson's idea is, to take into account the motion of perceived agents, to make a short projection in time and in space of the position of the perceived agents and not only a projection in time or in space of the current agent itself. It would improve many models dedicated to locomotion providing that the concepts developed by Gibson should be implemented.

AGENT MODEL FOR THE LOCOMOTION

The proposed model in Figure 2 is partially based on the well known schema "perception/decision/action". Furthermore the model is also based on in one hand on the Archisim model, a road traffic simulation model developed for more than 10 years at Inrets [Champion et al. 2001], [Auberlet et al. 2007] , and more generally on the models allowing the displacements of agents, and in the other hand on the literature dealing with the perception processes in psychology [Endsley 1995] and computer science.

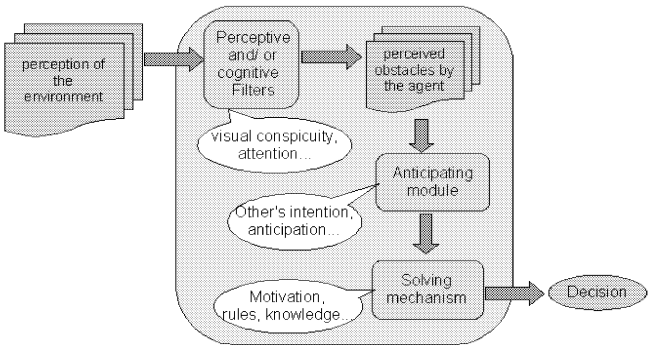


Fig. 2: architecture for an agent's decision model for the agent's locomotion

This model is based on the fact that a displacement could be seen as a succession of situations in time. The obstacles (other moving agent and/or objects) are seen as potential conflicts that are necessary to solve. Three main modules compose the model. The first is called "perceptive module". It consists in to define what the agent really perceives. Indeed, in some situations an agent can not perceive all what it sees. For example, it is not rarely to observe a driver or a pedestrian says "Oh! I did not see him!" whereas nothing hides the situation. The second module called "anticipating module" is related to the abilities of the agent to predict the intentions of the others. Finally, the last module called "solving mechanism" allows to the agent to take a decision after having to solve the existing conflicts. The works achieved by Gibsons should allow to improve the "anticipating module".

This agent model could be adapted to the displacement of autonomous agents in road traffic applications. Indeed, in this kind of environment, the agent can be a pedestrian, a driver or a motorcyclist. According to the knowledge on the abilities of the modelled entity, the perception module should be adapted but the way whose the agent will anticipate will be similar.

PERSPECTIVES AND FUTURE WORKS

This architecture actually exists in the road traffic simulation tools Archisim. The perspectives of this contribution are multiple. Firstly, it will be to compare the behaviour of actual models, as those coming from the field force domain, depending on whether Gibson's concepts are implemented or not. Secondly, in order to define a realistic pedestrian displacement model, in collaboration with psychologists, it is necessary to identify the information processed by a pedestrian during his displacement, and what he does with it. This work will strongly help us to model the two first modules (perception and anticipation) of the proposed agent's model.

ACKNOWLEDGMENTS

The research presented in this paper was performed as part of the French National project SARI-VIZIR sponsored by the National Research Agency.

REFERENCES

Auberlet J.-M.; Champion A.; Mandiau R.; Espié S. and Kolski C. 2007. "From perception to conflict resolution: a priority managing mechanism for road intersection. First results". In *Proceedings of the 8 6 th TRB Annual Meeting. CD-ROM.* Transportation Research Board of the National Academies Washington D.C., USA, 2007

Champion A.; Espié S. and Auberlet J.-M. 2001. "Behavioral road traffic simulation with ARCHISIM", In *Proceedings of Summer Computer Simulation Conference*, Orlando, USA, July 15-19, 2001.

Endsley M.R. 1995. "Toward a Theory of Situation Awareness in Dynamic Systems." Human Factors, 37(1), 32-64

Gibson, J. J., & Crooks, L. E. (1938). "A theoretical field-analysis of automobile-driving". The American journal of psychology, LI, 453-471.

Jean-Michel Auberlet is Research Scientist at the INRETS and is a member of the Modelling, simulation and driving simulators (MSIS) Department. He holds B.S. and M.S. degrees in Mathematics and a PhD in Biomathematics. Dr Auberlet's research interests includes model design of complex systems for simulation in various domains as in scheduling problem, physiology or as currently in psychology and road traffic.

A MODEL OF PROSPECTIVE MEMORY AND HABIT PHENOMENA CALIBRATED WITH DYNAMIC FIELD DATA

Robert Tobias, Christian Würzebesser and Hans-Joachim Mosler
Department of System Analysis, Integrated Assessment and Modelling (SIAM)
EAWAG: Swiss Federal Institute of Aquatic Science and Technology
Ueberlandstrasse 133, P.O. Box 611
CH-8600 Duebendorf
Switzerland
E-mail: robert.tobias@eawag.ch

KEYWORDS

Psychology, Numerical Modeling, Agent Based Simulation, Validation.

ABSTRACT

A formal model for investigating phenomena relating to forgetting the execution of intended behaviors in real-world settings is developed on the basis of literature findings. The model considers three events that raise accessibility (i.e. ease of remembering): the execution of the behavior, planning and preparing reminders (memory aids), and their effect. It also considers that the development of habits backs up memory tasks. The model takes into account that habit development is retarded by random fluctuations of the behavior performance and that habits are also built up for actions similar to those executed.
A first empirical test is presented using time-series data gathered during a campaign in Santiago de Cuba promoting solid-waste recycling. Significant behavior changes resulted by only distributing simple reminders to the households. As several households showed characteristic dynamics in their behavior change, the model could be calibrated to the data and replicated the observed tendencies in the dynamics. Further tests of the model are discussed and a general guideline for developing models based on psychological theories for application in decision support and campaign planning is presented.

INTRODUCTION

This article looks at prospective memory phenomena (for an introduction see Ellis and Kvavilashvili 2000); i.e. the problem of forgetting to execute an intended behavior at the critical moment and its resolution. Although this is a rather neglected area of psychological research, quite a lot of investigations have been devoted to it, as the problem is rather frequent in the case of simple but important everyday behaviors such as taking medicines (Sheeran and Orbell 1999) or shopping for the right products (Shapiro and Krishnan 1999). Forgetting to perform intended behaviors can even jeopardize campaigns of attitude change. On the other hand, people often have positive attitudes towards a behavior but do not perform it sufficiently often because they forget it. In such cases, measures that help them remember to perform the behavior even can be used as behavior change interventions. As such measures are much simpler and easier to spread in a population, they can be a very efficient way of promoting a desired behavior. But although reminders are helpful in practice, it is a challenging task to investigate their psychological mode of functioning. Prospective memory research is mainly pursued in laboratories, even though the relevant phenomena depend greatly on situational factors. It would consequently be more informative to investigate these phenomena precisely where reminders are used in real life, which would also allow the long-term effects to be investigated. But the problem then arises that most processes relating to prospective memory phenomena are unconscious and direct questions cannot be asked about them. The modeling approach presented here allows this dilemma to be resolved: findings from laboratory research about the unconscious processes can be used to define a specific model which is then used to explain time-series data of overt behavior in the field over longer time spans. Besides investigating prospective memory phenomena, therefore, this article also shows how the simulation method can bridge the gaps between laboratory and field data as well as between theory and application that exist in many areas of the social sciences.

FINDINGS TO PROSPECTIVE MEMORY

The investigation of memory phenomena is one of the oldest areas of experimental psychology (e.g. Ebbinghaus 1885). In brief, the findings point to the obvious fact that as more time goes by, the more difficult it can become to remember something. The ease of remembering can be modeled with the construct of accessibility that decays over time (e.g. Higgins 1996). Surprisingly, the investigation of

memory effects on behavior execution – known as *prospective memory research* (see Ellis and Kvavilashvili 2000 for an introduction) – started rather late. This research area focuses on the identification of factors that facilitate or inhibit recall of the behavior execution at the right moment. The model developed here considers only the literature dealing with the effects of situational cues, especially external memory aids such as notes or placement of a marker as a reminder (Intons-Peterson and Fournier 1986). The literature can be summarized briefly as follows: reminders activate associative connections between key situations and behaviors to be executed. This makes it easier to remember the behavior at the critical moment (e.g. Guynn et al. 1998). Moreover, artificial memory aids in particular back up retrospective memory tasks, i.e. help to recall details of how the behavior is to be executed (e.g. Shapiro and Krishnan 1999).

Not only do behavior options have different accessibilities, a specific behavior's accessibility to recall can also vary. Research on this aspect has focused mostly on the *cognitive resources* a person is able and willing to invest in the memory task. Recent studies indicate that prospective memory tasks always need cognitive resources (e.g. Marsh et al. 2002) although their extent may well vary (McDaniels and Einstein 2000).

So far, only the moment when a certain behavior is to be performed was considered. However, the moment when the future execution of a behavior is prepared is also important and has been well investigated. Both setting up a reminder (Goschke and Kuhl 1993; Intons-Peterson and Fournier 1986) and planning the execution of a behavior (Guynn et al. 1998) strongly increase the accessibility of the behavior to be executed in the future. All these aspects will be referred to in this article as the *effects of the intervention event*. It was shown that these effects become stronger as the preparation activities become more extensive (Rise, Thompson & Verplanken, 2003) and the commitment to the execution of the behavior becomes stronger (Gollwitzer 1999).

This sums up the main findings of prospective memory research. Nevertheless, before modeling can be started, it must be considered that this research is carried out almost entirely in laboratories. Accordingly, *characteristics of real-world settings* that differ from the laboratory settings must be added to the model. Three aspects seem relevant here: in real-world settings, behaviors are (1) frequently shown (2) over a longer period of time, and (3) their execution as well as situational cues which serve as reminders compete with other behaviors and cues. The *frequency of behavior execution* can facilitate or hinder remembering. Each action can be seen as an event that increases the accessibility of the behavior. On the other hand, however, the more often a behavior is performed, the more might this happen in situations and moments where remembering is more difficult. So, whether higher frequencies of behavior execution facilitate or hinder remembering depends on the time lag between individual execution events (the longer the time, the more difficult it is to remember higher behavior intensities) and the circumstances in which the behavior is to be performed.

The aspect of the long time span during which a behavior is performed leads to a completely different area of psychological research, namely *habits*. A habit is built up by performing a behavior repeatedly in similar situations (Oulette and Wood 1998). The associative connections between a situation and a behavior facilitate its recall in the situation in which it is often performed (Aarts et al. 1998). In other words, habits do not increase the accessibility of behavior options but lower the cognitive load required to remember those options with lesser accessibility. Almost no research has been done on the development of habits over time. The closest approach has come from behaviorist experiments on enforcement learning. They showed that the gain in learning a behavior was fast at first and then slowed down to reach an optimum after about 14 days (Tolman and Honzik 1930). The findings relating to behavior shaping (Skinner 1953; 1958) that show a rather general formation of associations are very important. They suggest that habits not only build up for actions that are actually executed but also for similar actions in similar situations. Finally, there exists an implicit consensus that habits decay over time if the behavior is not executed.

Finally, in contrast to laboratory settings, in the real word many situational cues interfere with a reminder and many behaviors compete for performance. This aspect is difficult to tackle. The main point seems to be how important the performance of the behavior is for a person. If a person feels committed to perform a behavior, a reminder will have stronger effects despite distracting cues and competing behaviors. This aspect has been intensively investigated in the area of *implementation intentions* (e.g. Gollwitzer 1999): thus the activation of the association between a situational cue and a behavior also depends on how strongly a person feels committed to the performance of the behavior (e.g. Chasteen et al. 2001; Orbell et al. 1997).

This briefly summarizes the findings that have to be considered in a model of prospective memory. The next step is to formalize and integrate the findings and complete them with further assumptions.

THE MODEL

The model is agent-based where each agent represents one person. It is also deterministic for reasons of complexity reduction. The behavior is modeled in discrete intensities each of which has its own attributes, such as the habit set up to perform it. Each of these intensities represents a set of single actions that are executed in certain situations. It is assumed that all actions of lesser intensity are also executed in the case of higher intensities even though a higher intensity means that additional actions are performed. How many intensities are modeled depends on the number of *psychologically* different actions and situations of action-execution to be considered for a behavior. For the simulations presented here, 21 behavior intensities were used.

A model of prospective memory will now be set up on the basis of the presented findings. A schematic overview of the model is given in Figure 1.

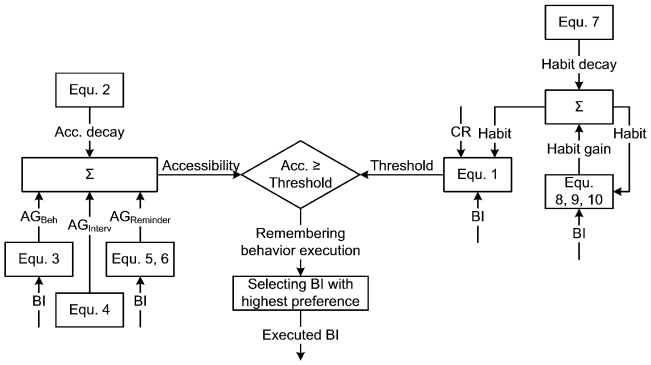


Figure 1: Schematic Diagram of the Model

The basic idea is that remembering a certain behavior option can be more or less easy, i.e. a behavior may be more or less accessible, but how accessible it must be to be remembered depends in turn on other factors. The rhombus at the center of Figure 1 symbolizes that if the accessibility is greater or equal to a threshold value, a behavior option is remembered, otherwise not. This comparison is performed for each behavior intensity modeled. Of all remembered behavior intensities, the one with the highest preference is selected. It must be emphasized that the preferences (i.e. attitudes) are kept constant over time for the investigations presented here. The findings presented above suggest that accessibility decays over time and is increased by several events such as the execution of the behavior (AG_{Beh}, where AG stands for the accessibility gain), the intervention event (AG_{Interv}) and the effect of reminders ($AG_{Reminder}$). The accessibility threshold is increased by the behavior intensity (BI), as higher intensities of behavior execution mean that the behavior might be performed with increasing frequency at moments when remembering is more difficult. The threshold is decreased by the cognitive resources used for remembering (CR) and the habits developed for the behavior. Habits themselves decay over time and are increased by the execution of the behavior. The blocks contain references to the equations that will now be presented and explained.

A Formal Model of Prospective Memory

On the basis of the findings presented, the *accessibility threshold* for a behavior intensity BI is calculated as follows:

$$Threshold_{BI} = C_{Rem} - WCR_{Rem} \times CR - WH_{Rem} \times habit(BI) + WBI_{Rem} \times BI^{PBI_{Rem}} \quad (1)$$

The threshold is composed of a constant C_{Rem} that is reduced by the influence of cognitive resources CR and the *habit* of executing the behavior with an intensity BI. Each of these influences has its own weight (WCR_{Rem} for CR and WH_{Rem} for $habit_{BI}$). As explained above, the threshold is increased by the behavior intensity. This influence again has its own weight (WBI_{Rem}) and is modeled here as a power function (PBI_{Rem}). This allows us to model whether the greatest increase in difficulty of remembering is at a low ($PBI_{Rem} < 1$) or a high ($PBI_{Rem} > 1$) behavior intensity. Nevertheless, other functions might be used to model specific circumstances of behavior execution. The cognitive resources CR are set constant over time for this investigation. The development of habits will be examined later on. The current focus will therefore be on modeling the dynamics of accessibility.

Findings on the *decay of accessibilities* point to a relatively fast decay at the beginning and a slower decay later on. Such dynamics can be modeled simply and independently of time with a proportional decay. The accessibility $Acc(t)$ can therefore be calculated as follows:

$$Acc(t) = Acc(t-1) - Acc(t-1) \times ADP + AG_{Beh}(t) + AG_{Interv}(t) + AG_{Reminder}(t) \quad (2)$$

In every time-step, the accessibility is lowered by a fraction of its old value. So when the accessibility is high, decay is fast and slows down as the accessibility is reduced. The general speed of decay is set by the accessibility-decay parameter ADP. In a time step accessibility not only decays but also can increase. These accessibility gain (AG) terms will be explained next.

Of the three events that increase accessibility, the *behavior execution* (AG_{Beh}) will be presented first of all. The findings state that the strength of the effect should increase in line with the behavior intensity, as the latter models the frequency of behavior execution. The same function ($WBI_{Rem} \times BI_{Exe}^{PBI}$) is used as in the accessibility threshold (Equation 1). This is because the actions and situations can be assumed to be similar for the investigations presented here. In both cases, therefore, the same frequency effect is in principle modeled. Nevertheless, if more detailed information is available, a different function may be used here. To take into account the fact that the increase in accessibility and in the difficulty of recall due to higher behavior intensities may be different, the formula is suitably weighted by the addition of (WBI_{AGBeh}). Depending on the time lag between single execution events and the circumstances in which the behavior is to be performed, frequent behavior execution either facilitates remembering ($WBI_{AGBeh} > ADP$) or hinders it ($WBI_{AGBeh} < ADP$). In conclusion, depending on the intensity of the *executed* behavior (BI_{Exe}), the accessibilities of all behavior intensities are increased by the same amount of AG_{Beh} defined as:

$$AG_{Beh} = WBI_{AGBeh} \times WBI_{Rem} \times BI_{Exe}^{PBI_{Rem}} \quad (3)$$

The *effect of the intervention event* (AG_{Interv}) is modeled very simply: it is assumed that the accessibility increase is directly proportional to the magnitude of the preparation activities and the commitment to the execution of the behavior. This is expressed by the construct of commitment intensity:

$$AG_{Interv} = CommitmentIntensity \quad (4)$$

The final event which increases the accessibility is the *effect of a reminder* ($AG_{Reminder}$) at the moment when the behavior is to be performed. The findings suggest that three influences on this effect must be considered: the characteristics of the reminder, the intensity of the commitment and the characteristics of the situation. The first two aspects are defined as parameters ($AGP_{Reminder}$ and *CommitmentIntensity*) and the third as a function (*Sf*) and they are assumed to interact multiplicatively, i.e. if one of these factors is zero, the others cannot compensate it.

$$AG_{Reminder} = AGP_{Reminder} \times CommitmentIntensity \times Sf(BI) \quad (5)$$

What makes modeling the effect of reminders difficult is the consideration of situational influences. As explained above, the behavior intensity is an abstraction from many single behavior-execution events. Each of these events may take place in quite a different situation. It must in particular be considered that the behavior is not executed close to the reminder in all situations. So *Sf(BI)* must express how similar the different situations are and it is therefore called the *similarity function*. Even though the situational conditions may be different for each case investigated and should be clarified before applying the model, a rough approximation to a larger range of cases is attempted here. It is assumed that most single behaviors are performed close to the reminder up to a certain behavior intensity. As this increases, the behavior may be performed an increasing distance from the reminder so that the latter has less and less effect. For very high behavior intensities, however, there should not be much difference in the reminder effect. To describe this characteristic, a logistic function is used as follows:

$$Sf(BI) = \left(1 - \frac{\frac{1}{1+e^{(0.5-BI^{T_S})\times S_S}} - \frac{1}{1+e^{0.5\times S_S}}}{\frac{1}{1+e^{-0.5\times S_S}} - \frac{1}{1+e^{0.5\times S_S}}} \times DP \right) \quad (6)$$

The core of Equation (6) is a logistic function of *BI* with a turning-point parameter T_S and a slope parameter S_S. A linear transformation assures that the function runs through 0 and 1. The function is further multiplied by the dissimilarity parameter *DP*. The similarity function *Sf(BI)* defines how similar two behavior intensities *BI* are: the smaller the difference of the similarity function for two behavior intensities, the more similar are their situations. The situation in which the reminder has the maximum effect has the value 1. This value becomes smaller as a function of how greatly the situations of the behavior-execution events of a certain intensity differ from this standard situation. The similarity function has the shape of a step: Up to a certain intensity, the value of the function changes very little, thus indicating rather similar situations for these behavior intensities. The value then decreases rapidly, indicating that actions with higher behavior intensities are performed in rather different situations, particularly as they move further away from the reminder. Then again, the value is nearly constant, which means that the situations are similar to one another although dissimilar to the situations with lower behavior intensities. The turning-point parameter T_S determines at which intensities the step of the dissimilarity function is located; the slope parameter S_S sets the width of the step and the dissimilarity parameter *DP* its height.

This treatment formalizes most of the findings described above and defines the actual model of prospective memory. However, one important aspect is still missing, namely the development of habits. This model part will now be presented.

Modeling the Development of Habits

It has already been mentioned that research into the development of habits is almost nonexistent. Most formulations in this part of the model must consequently be based on assumptions. The same function is assumed for the *decay of habits* as for the decay of accessibilities, even though habits may decay at different speeds than accessibilities: this is defined by the habit decay parameter *HDP*: As Equation (2) the habit strength *habit(t)* can be calculated as follows:

$$habit(t) = habit(t-1) - habit(t-1) \times HDP + habitGain(t) \quad (7)$$

For the formalization of the *habit gain,* random fluctuations of the behavior intensity must be considered. Although the model is deterministic and calculates a behavior intensity trend, what counts for the habit development is the actually executed behavior and not its trend. In a deterministic model this problem cannot be solved just by adding some random noise. The fluctuations must be explicitly estimated and their effect defined. To simplify the model only the effect of sometimes not showing this behavior at all is considered. It is assumed that the habit increase is directly proportional to the frequency of behavior executions. So if the behavior is only performed every second day, for example, the habit increases half as much as it would if it were performed daily. In order to estimate the frequency of a behavior not being executed at all in reality due to random events – even though the model, which abstracts from these random events, calculates a behavior intensity (BI_{Exe}) greater than zero – two influences are considered: habits and the value of the deterministic trend of the behavior intensity (BI_{Exe}).

It is assumed that if remembering to execute a behavior is based mainly on habits, it is less sensitive to random perturbations than if it were based mainly on accessibility. This assumption is correct if accessibilities change much faster than habits. Therefore, one element of the estimation of behavior fluctuations is that the stronger the habits the less probable it is for the behavior not to be performed at all on a particular day. For parsimony, a linear relation is assumed here. Nevertheless, it seems insufficient to consider only the habit for the executed behavior (BI_{Exe}) according to the model. If habits for lower behavior intensities are stronger, they might prevent the behavior from not being executed at all. Therefore the strongest habit (*maxHabit*) of all behavior intensities greater than 0 and less than or equal to BI_{Exe} is used in the equation. So the habit increase is proportional to *maxHabit*.

Moreover, the higher the behavior intensity actually executed (BI_{Exe}) according to the model, the lower the probabil-

ity that the behavior will not be performed at all in reality due to random perturbations. It is assumed that the effects of such perturbations are normally distributed. So if BI_{Exe} is very low, there is a high probability that the behavior was not shown at all in reality. The probability of not showing the behavior at all should decrease rapidly with increasing BI_{Exe} and would already be quite low when this was at a medium level. A logistic function is used to express this tendency in a closed formula. So the probability of a behavior not being shown at all due to random perturbations, despite the model calculating a behavior intensity BI_{Exe} greater than 0, is calculated as follows:

$$p0f(BI_{Exe}) = \frac{\frac{1}{1+e^{\left(0.5-BI_{Exe}^{T_H}\right)\times S_H}} - \frac{1}{1+e^{0.5\times S_H}}}{\frac{1}{1+e^{-0.5\times S_H}} - \frac{1}{1+e^{0.5\times S_H}}} \quad (8)$$

The turning-point (T_H) and slope parameter (S_H) can be interpreted identically as in Equation (6), although in this case these parameters define the step in the speed of habit development. For intensities below this step, habits develop slowly, while above the step they develop much quicker. It must be noted that these parameters cannot be set as freely as for Equation (6), as this formula aims at imitating the effects of normally distributed random perturbations on the behavior intensity. The habit increase for the executed behavior (BI_{Exe}) according to the model is then calculated as follows:

$$habitGain_{Exe} = \left(maxHabit \times \left(1 - p0f(BI_{Exe})\right) + p0f(BI_{Exe})\right) \times HDP \quad (9)$$

Equation (9) formalizes the tendency for the habit to develop faster with greater habit (*maxHabit*) and an increased trend of executed behavior intensities (BI_{Exe}). The habits (*maxHabit*) have a linear effect while the influence of the behavior intensity (BI_{Exe}) matches a logistic function (*p0f*) that represents the normal distribution of random perturbations on the behavior execution. As habits can only compensate random perturbations that actually occur, *maxHabit* is multiplied by ($1 - p0f(BI_{Exe})$). The maximum speed is the same as for the decay of habits (*HDP*). So after infinite execution of a behavior at certain intensity, its habit reaches the maximum of 1.

Equation (9) defines the increase in the habits of executed behavior intensities and thus of executed actions in certain situations. As explained above, behavior-shaping findings indicate that habits develop not only for actions that are performed but also for similar actions in similar situations. So the more similar that actions and situations are for a higher than executed behavior intensity, the more habits will also build up for this behavior intensity. For the sake of simplicity, it is assumed here that the actions for all behavior intensities are the same. The dissimilarity function of the situations as expressed in Equation (6) can consequently be used here again. The habits of the intensities of an executed behavior then increase by:

$$habitGain_{BI} = \min\left(1, \max\left(0, Sf(BI) + \left(1 - Sf(BI_{Exe})\right)\right)\right) \times habitGain_{Exe} \quad (10)$$

For behavior intensities *BI* lower than those executed, the habits increase at the same rate as for the executed intensity BI_{Exe} by $habitGain_{Exe}$ (Equation 9). For greater intensities *BI*, the rate of habit increase is proportional to the similarity of the situations to that of the executed intensity BI_{Exe}. This is determined on the basis of the similarity function *Sf* (Equation 5).

EMPIRICAL INVESTIGATION

Campaign, Data Gathering and Methods of Analyses

The data was gathered in the pilot phase of a *campaign* to reduce deposited solid waste in Santiago de Cuba at the beginning of 2005. Foregoing investigations indicated (Mosler et al. in press) that attitudes to recycling were very positive but forgetting to perform the relevant behavior was mentioned as a problem. A simple intervention based on reminders was consequently implemented: every household in the sample received a sheet of paper with the following text in Spanish: “Please classify and separate! Glass – Aluminum – Paper – Cardboard – Plastic”. Without further persuasion, the paper was handed out during the first interview with the request to hang it up in the house where most of the waste is collected.

For the *analyses* of the model, empirical data must be aggregated to reduce the random fluctuations of the behavior intensity. To avoid different dynamics being mixed up in an aggregation, the types of dynamics were defined in three steps using different forms of aggregations and tests: first, the characteristics and types of the dynamics of the behavior intensity were determined visually. Cross-tabulation and variance analyses were then performed with data aggregated over four measurements in each case at four points in time. These analyses were used to classify the dynamics quantitatively and to test the significance of the classification. Finally, means of the time-series were calculated for each type of dynamics found. These time-series were then used to calibrate the model. As the empirical data was collected daily, the simulation was set up such that one step represents one day. Although data is only available for 27 days, the simulation ran for 50 steps. The first steps show the stable start-setting of the model. The intervention takes effect on the 10th step and the comparison with the data starts at step 11.

The visual and statistical analyses resulted in four *types of dynamics* of the behavior intensity. Type 1 (n = 14) shows a constant high behavior intensity after the intervention. Type 2 (n = 5) starts with a high behavior intensity on the day of the intervention; the behavior intensity then decreases sharply but increases again later on. After reaching the intensity of the intervention day, the behavior intensity stays constant. Type 3 (n = 4) also starts with a high behavior intensity on the day of the intervention and then decreases sharply. Unlike Type 2, however, the behavior intensity

does not increase again but decreases slowly to zero. Finally, Type 4 (n = 7) shows a constant low behavior intensity. Assuming a lower behavior intensity before the campaign, these types can also be interpreted with respect to the success of the interventions. Type 1 would indicate complete success, while Type 2 could be interpreted as successful with some starting problems. The other two types stand for failed interventions, although in the case of Type 3 the intervention was successful in the short term. As Types 1 and 4 do not show any dynamics in the development of the behavior intensity only investigations of Types 2 and 3 will be presented here.

Calibrating the Model to the Empirical Data

The simulation was calibrated to the data of the four types. The parameter settings for all types are equal except for the commitment intensity, the start behavior and the start accessibility. In Figure 2, the resulting simulated developments of behavior intensity (dashed lines) are compared with the empirical time series (solid lines) for Types 2 (above) and 3 (below).

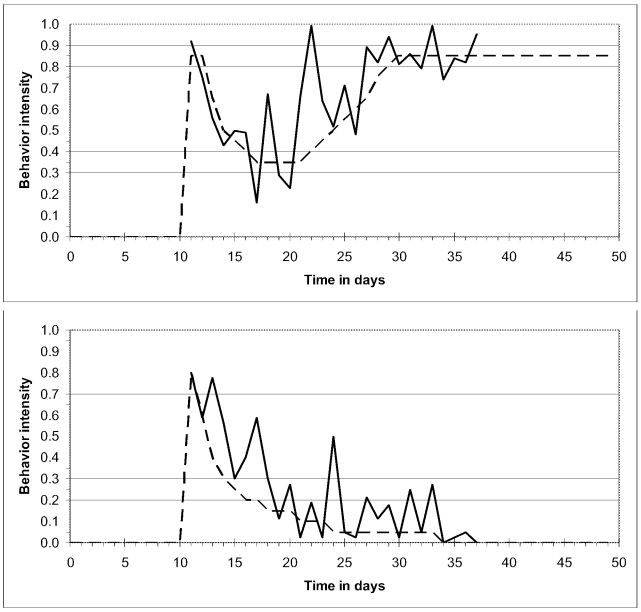


Figure 2: Calibration Result for Type 2 and 3

The good visual fit of the simulation to the empirical data in Figure 4 is also reflected in the *quantitative fit parameters*. Although the largest absolute error is 0.59 for Type 2 and 0.45 for Type 3, in only three and four cases respectively is the deviation from the empirical data greater than 0.25 – the resolution of the measurement. The average absolute deviation per day is 0.13 and 0.12 respectively. Moreover, the simulation explains 50% of the variation for Type 2 and 65% for Type 3. The fit is perturbed mainly by spikes at days 18, 22, 27, 33 and 37 for Type 2 and at days 17, 24 and 31 for Type 3. As theses spikes follow a roughly weekly pattern, it is assumed that the interviewers collected the questionnaires and ran the short interview more often on those days. These events then reminded the persons to perform the behavior, thus increasing the behavior intensity. The effects are expected to be especially strong when the behavior intensity is low despite the accessibility or habits being relatively high. If the behavior intensity is high, nothing needs to be remembered and if accessibility and habits are very low, reminders will not have much effect. Accordingly, the strongest impact can be found for Type 2 on day 22 and for Type 3 on day 24. As these events are not modeled in the simulation, these days should be considered especially when testing the fit of the simulation to the empirical time series. In fact, the results do get much better if the days with assumed interviewer visits are omitted from the fit calculation. The maximal absolute deviation is reduced to 0.31 and 0.37 for Types 2 and 3 respectively. Besides these values, only one further value for Type 3 has an absolute deviation of more than 0.25. The mean absolute error per day is 0.10 for Type 2 and 0.09 for Type 3. Finally, the simulation now explains 72% and 78% of the variation in the behavior intensity of Types 2 and 3 respectively.

As a first conclusion, it can be said that the model passed the first test. It could be calibrated to a specific case and the fit with the data was good.

DISCUSSION

This article presents a model for investigating the effects of forgetting to perform an intended behavior. The basic idea of the *model* is that it can be more or less difficult to remember the behavior performance and that a variable threshold defines whether the accessibility of a set of actions is sufficiently high to be remembered and therefore executed. While accessibility decays over time and is increased by certain events such as the execution of the behavior or the effect of a reminder, the threshold depends mainly on the cognitive resources used for the memory task and habits developed for the various actions in relevant situations. Habits also decay over time and increase when the behavior is performed. The model considers that the development of habits is retarded by fluctuations in behavior performance due to random events and that habits develop not only for the actions performed but also for similar actions in similar situations. The model was applied to explain the time series of behavior dynamics in a real behavior-change campaign: it replicated the empirical data well and could explain the causes and determinants of the observed dynamics. It consequently appears to be well suited for investigating real-world phenomena caused by forgetting to perform an intended behavior.

The presented test is only a small part of a much larger investigation of the model, which can not be presented here in detail. Nevertheless it shall be mentioned, that the model was also calibrated and tested with data from a campaign in Bolivia promoting water disinfection and using a completely different measure for behavior performance. The mentioned types of dynamics were confirmed and the model fitted even better than to the presented data even without recalibration. With a slight adjustment the simulation explains here 98% of the variance of type 2 and 87% of

type 3. Therefore the results can not be ascribed to the type of behavior, the measurement or cultural influences. It also indicates a general applicability of the model at least to a certain extend.

Further, system analytical investigations like identification, sensitivity, and feature analyses are being performed. First results indicate a reasonable sensitivity of all relevant parameters reconfirming the model design. Even though an infinite number of parameter settings lead to similar good fits as presented, the value of most parameters can be changed only by a quarter of their range or less to be compensated by changes of other parameters – an acceptable result considering the quality of the data.

Finally, from systematic parameter variations the dynamics observed in the data could be explained by the model as follows: The intervention event raised accessibility of the behavior, which decayed within few days leading to forgetting the behavior performance more and more often. This is manifested in the decay of behavior intensity as observable in Figure 2. The slow down of this decay and the increase of the behavior intensity for type 2 can be attributed to the development of habits after performing the behavior for a week or two. The reason for the difference of the development for type 2 and 3 lies primarily in the lower commitment to perform the behavior for the latter. Although the difference to Type 2 is small, it is catalyzed by situational characteristics and random perturbations that lead to a massive slowdown of habit development when the intensity drops below about 0.3.

Even though this article focuses on a model for investigating prospective memory and habit phenomena, it can also be understood as an example for a modeling approach known as *theory-based application-oriented modeling*. It designs rather complex agents in a way that allows all parameters to be related to concrete and practically important real-world characteristics. This makes it easy to incorporate a broad range of findings from laboratory and field research, to fill research gaps with additional hypotheses and to relate the model to specific real-world cases. A further advantage of this approach is that it goes beyond a mere investigation of general principles at abstract level to include concrete real-world dynamics. Due to the specificity of the models, they can represent findings from laboratory experiments as well as from field surveys. Moreover, the explicit representation of factors of practical importance even allows behavior-change campaigns to be modeled. Such models are developed in five steps as already presented in this article.

1. *Defining the phenomenon to be modeled*: the article already started with a predefined phenomenon, i.e. forgetting the performance of an intended behavior. It is important to define the phenomenon from an application-oriented perspective and not to frame it within a single research tradition.
2. *Compiling the available findings*: those research areas that could add important findings and theories must be determined on the basis of defining the phenomenon. Most application-oriented problems are related to many different research areas, although in most cases only to parts of them.
3. *Integrating and formalizing the findings*: probably the most difficult step is to get the many findings and theories to work together. The definition of constructs and functions requires some abstraction, and the findings and theories must be reduced to their key statements. Moreover, research gaps must be discovered and filled with hypotheses. It is important to model whatever is necessary for explaining the investigated phenomena and not what is available in the literature!
4. *Testing the model*: any model to be used for investigating real-world phenomena must be tested with real-world data. The empirical data must therefore be similar to the artificial data. For example, it would not be satisfactory to test a model that produces time series of individual behavior with aggregated data of a population before and after a campaign, but it would be better than no test at all. Moreover, the quality of a model depends not only on its fit to the data but even more strongly on its ability to explain the observed dynamics.
5. *Application of the model*: even though the aim of this modeling approach is to develop aids for planning and conducting behavior-change campaigns, no model ever can make decisions. A model such as the one presented here can only be one tool among others designed to support those in charge to reach good decisions quickly or at least to avoid the bad ones. In particular, a model is always only an abstraction of the world and any findings of simulation research must be set in relation to the real world by an expert.

In this article it could be shown that the main parts of the modeling approach work: a large set of findings was integrated and formalized, research gaps were identified and filled with explicitly stated assumptions and real-world data was replicated and explained. So the model can at least explain how memory aids in real-world settings work, which was the principal aim of this article. Nevertheless, further tests with more data are necessary to start using the model as a help for designing campaigns.

CONCLUSION

Investigating psychological phenomena in real-world settings is as challenging as important. Societal problems happen in the real world and not in laboratories. However, psychological processes often can not be investigated directly. Analyzing dynamic data, which can be gathered more or less easily in the field, using computer models based on laboratory findings might solve this dilemma. This article presented a successful application of such an approach and some examples of findings derived from it, like long term dynamics or interactions of psychological processes traditionally assumed as unconnected. Models like the presented might be criticized as oversimplifying and at the same time overly complex. However, the example illustrates, that the model could replicate and explain an empirical phenomena based on psychological findings. System analytical investi-

gations running at the moment also show that the model can be handled analytically. Depending on the task that shall be performed with the model, it might be simplified or expanded, but for the case presented it worked fine.

It can be concluded that theory-based and application-oriented modeling offers a promising approach to overcoming many shortcomings in current social scientific research. It particularly allows the gaps between laboratory and field research as well as between theory and application to be bridged. Models such as the one presented here allow valuable findings of laboratory research to be used for solving real-world problems. Computer simulation may consequently become an important tool for tackling societal challenges.

REFERENCES

Aarts, H.; B. Verplanken; and A. van Knippenberg. 1998. "Predicting behavior from actions in the past: Repeated decision making or a matter of habit?" *Journal of Applied Social Psychology, 28*, 1355–1374.

Chasteen, A.L.; D.C. Park; and N. Schwarz. 2001. "Implementation intentions and facilitation of prospective memory." *Psychological Science, 12*, 457–461.

Ebbinghaus, H. 1885. *Über das Gedächtnis*. [About the memory.] Leipzig: Duncker.

Einstein, G.O. and M.A. McDaniel. 1990. "Normal aging and prospective memory." *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 717–726.

Ellis, J. and L. Kvavilashvili. 2000. "Prospective Memory in 2000: Past, Present, and Future Directions." *Applied Cognitive Psychology, 14*, S1–S9.

Gollwitzer, P.M. 1999. "Implementation intentions. Strong effects of simple plans." *American Psychologist, 54*, 493–503.

Goschke, T. and J. Kuhl. 1993. "Representation of intentions: Persisting activation in memory." *Journal of Experimental Psychology: Learning, Memory and Cognition, 19*, 1211–1226.

Guynn, M.J.; M.A. McDaniel; and G.O. Einstein. 1998. "Prospective memory: When reminders fail." *Memory and Cognition, 26*, 287–298.

Higgins, E.T. 1996. "Knowledge activation: Accessibility, applicability, and salience." In *Social psychology. Handbook of basic principles*, E.T. Higgins and A.W. Kruglanski (Eds.). Guilford, New York, 133–168.

Intons-Peterson, M.J. and J. Fournier. 1986. "External and internal memory aids: When and how often do we use them?" *Journal of Experimental Psychology: General, 115*, 267–280.

Marsh, R.L.; T.W. Hancock; and J.L. Hicks. 2002. "The demands of an ongoing activity influence the success of event-based prospective memory." *Psychonomic Bulletin and Review, 9*, 604–610.

McDaniels, M.A. and G.O. Einstein. 2000. "Strategic and automatic processes in prospective memory retrieval: A multiprocess framework." *Applied Cognitive Psychology, 14*, S127–S144.

Mosler, H.-J.; A. Tamas; R. Tobias; T. Caballero Rodríguez; and O. Guzmán Miranda. in press. "Deriving interventions on the basis of factors influencing behavioral intentions for waste recycling, composting and reuse in Cuba." *Environment and Behavior*.

Orbell, S.; S. Hodgkins; and P. Sheeran. 1997. "Implementation intentions and the theory of planned behavior." *Personality and Social Psychology Bulletin, 23*, 945–954.

Ouelette, J.A. and W. Wood. 1998. "Habit and intention in everyday life: The multiple processes by which past behavior predicts future behavior." *Psychological Bulletin, 124*, 54–74.

Rise, J.; M. Thompson; and B. Verplanken. 2003. "Measuring implementation intentions in the context of the theory of planned behavior." *Scandinavian Journal of Psychology, 44*, 87–95.

Shapiro S. and H.S. Krishnan. 1999. "Consumer memory for intentions: A prospective memory perspective." *Journal of Experimental Psychology: Applied, 5*, 169–189.

Sheeran, P. and S. Orbell. 1999. "Implementation intentions and repeated behaviour: Augmenting the predictive validity of the theory of planned behaviour." *European Journal of Social Psychology, 29*, 349–369.

Skinner, B.F. 1953. *Science and human behavior*. Macmillan, New York.

Skinner, B.F. 1958. "Reinforcement today." *American Psychologist, 13*, 94–99.

Tolman, E.C. and C.H. Honzik. 1930. "‚Insight' in rats." *University of California Publications in Psychology, 4*, 215–232.

BIOGRAPHIES

ROBERT TOBIAS studied civil engineering at the Swiss Federal Institute of Technology in Zurich, where he gained his degree in 1989. After four years experience as a geotechnical engineer, he went on to study psychology, computer science and sociology at the University of Zurich, where he received his PhD in psychology in 2006. He is currently working on computer simulations of social-scientific theories and intervention measures for communities at the Swiss Federal Institute of Aquatic Science and Technology (Eawag).

CHRISTIAN WÜRZEBESSER studied computer science at the University of Passau, Germany. He is a scientific collaborator for social simulation at the research group "Modelling Social Systems" at Eawag. His current research is on modelling human behaviour and cognitive processes.

HANS-JOACHIM MOSLER is an associate professor of social and environmental psychology at the University of Zurich, Switzerland and head of the research group "Modelling Social Systems" at Eawag. His current research is on behavioural change in large populations using agent-based simulations.

Organisational Learning via Interactive Process Simulation in AGE

N. B. Szirbik
G. B. Roest
{n.b.szirbik, g.b.roest}@rug.nl

The Agent Laboratory
Faculty of Economics and Business, University of Groningen,
Landleven 5, PB 800, 9700 AV Groningen, The Netherlands.
http://www.agentlab.nl

KEYWORDS
Business Information Systems, Interactive Process-Oriented Simulation

ABSTRACT

In this paper, the concept of Interactive Process Simulation is introduced as a specialisation of Business Gaming. A specific gaming and agent development framework, based on interactive simulation and a specific modelling langauge, is shortly presented. The concepts of the language are explained via a simple example illustrating an interactive process in a logistic environment. Finally, the statement that process simulation leads to organisational learning is argumented.

INTRODUCTION

In the area of business process modelling, the analyst is in most situations facing the fact that a large part of knowledge is not explicit but tacit. According to Polyani, the tacit body of knowledge is characterized by the fact that it is personal, context specific, and therefore initially hard to formalize and also communicate (Polyani, 1966). The explicit body of knowledge, on the other hand, is typically easy to illustrate and communicate through a "systematic" (formal or quasi-formal) language. Polyani's empirical investigation reveals that managers and human beings in general accrue knowledge by consciously creating intentions and planning their own actions, based on previous experiences. His immediate corolary is that explicit knowledge represents only "the tip of the iceberg" of the entire body of knowledge of a person or of an organisation. In (Nonaka and Takeuchi, 1995) there is a dynamic model called the *knowledge conversion process*, based on the assumption that human knowledge is created and expanded through something they call "social interaction" between tacit and explicit knowledge.

Organisations are adapting nowadays from a functional based knowledge acquiring style towards a process based one. Fisher proposes a process maturity model (Fisher, 2004), and investigates how various organisations have moved from the so-called "silo knowledge organisation", towards a networked knowledge organisation. Such businesses strive towards explicit business process models, which enable the use of new technologies from the BPM (business process management) area, like interweaved ERP systems, workflow systems, webservice based systems, multi-agent systems (MAS), collaborative work support systems, etc. The ability to discover fast (learn) the process models (i.e. making them explicit instead of tacit), and also to change them easily and communicate the explicit knowledge to the actors in the organisation is considered by business analysts as "the main factor" to have competitive advantage for an enterprise. However, to make knowledge explicit, and build complex organisational processes (Daft, 2004) that are executed by actors with different viewpoints and experience is a very difficult task. Typically, business processes emerge and they can be optimised and fine-tuned to achieve the goals of the organisation over time. Another way, used by newly enacted organisations, is to use "best-practice" models from similar and successful organisations and adapt the organisation (top-down learning) in a way that suits the overall process models. In an abstract way, business processes can be visualised and dynamic presentations can convey to the actors the knowledge that can lead to mental beliefs consistent with the process structure and goals. Moreover, interactive simulations of the process, involving the actors who can take action and see the repercussion of their decisions, give the opportunity to identify exceptions, unforeseen circumstances, and also solutions for these (i.e. "what-if" scenarios). Our research focus is the third situation, where the actors know very well what to do, but they want to discover early (in a controlled environment and not by trial and error emergent practices) the "best" processes in a new organisation.

The main component of our research framework (Stuit and Szirbik 2007) is AGE (Agent Growing Environment). This is an interactive simulation environment used for research, education, and Multi-Agent System (MAS) development purposes at the University of Groningen. The AGE tools and theoretical aspects are developed mainly by the researchers and developers within TAL (The Agent Laboratory at the University of Groningen). The environment is used in a manner similarly to business gaming (Lainema, 2004), where human players can enact a business situation (e.g. supply chain, market, negotiations, auctions, contracting, etc). In AGE, instead of economic aspects (like costs/prices in existing market simulation) the focus is on the agent based social structure (Berry, 2002; Bonabeau, 2002) of the business processes within one or more organisations that are modelled in the agent-oriented language TALL (The Agent Lab Language). This modelling language allows on one side the players to grasp the structure and dynamics of the simulated organisations that they are "piloting", and also has

a partially-defined execution semantics (based on Behaviour Nets, see Meyer and Szirbik, 2007) that allows that many actions of the simulated items in the model can execute various actions themselves (similar to engaging the automatic pilot in a flight simulator). In this way, the players can focus on those aspects that are important for the simulation, and rest aside details that are routine, repetitive and can be easily delegated to the simulated agents.

This paper is organised as follows: the next section explains the agent related concepts of our framework, the third section presents and example illustrating the escape/intervention mechanism, and the last section concludes the paper.

BASIC CONCEPTS IN THE AGE FRAMEWORK

Besides the external players (and experimenters), from a modelling and simulation perspective, we distinguish between three types of agents in the AGE toolset: *human, software* and *synthetic*.

- *Human agents* represent the people that make up an organization.
- *Software agents* are computer applications that autonomously carry out certain activities of the business process on behalf of a human being or an entire department/organisation (inspired from Wooldridge, 2002).
- *Synthetic agents* are similar to institutional agents (Carley 2000; Wagner 2003) that are social constructs such as organizations and organizational units. In TALL, all agents types can be assigned to roles and interact with other agents. This mimics real-life settings in which for instance a patient is interacting with a hospital (a synthetic agent).

TALL is used mainly for modelling *interactional processes* and the TALL models drive the simulation, development and deployment of agents. An interactional process is one that can be reduced easily to a composition of interaction descriptions. TALL is part of a larger methodology that supports the capturing of interactions via simulation of the business process, involving human experts/stakeholders from the business domain under consideration. Simulation participants impose their behaviour in interactions on the agents they are piloting. In reality, when humans come across exceptional/unknown circumstances or events they usually invoke higher authority (Morgan 2006). The behaviour of the human participants of the simulation around these exceptions, supposedly can solve the problem their agents are presented with. We say that an agent is going into *escape mode* when its behaviour is insufficient or inappropriate for a particular interaction context. To solve an escape situation, the external players can change the behaviour of the agent in case (this behaviour is expressed in a graphical way, via one of the TALL diagram types – i.e. Behaviour Nets, see figures). We call this action *intervention.* Players can intervene also when they it consider necessary, even without an escape situation being triggered. New software agents are created by instantiation during the simulation sessions where the human agents act as expert players. Inside the simulation sessions, the software agents possess (limited/incomplete) behaviour descriptions. During each escape that occurs in the simulation sessions, the behaviour of the software agents is enriched with the logged behaviour of the humans (this process is called “growing” the agents). Each session will bring more automation to the simulation. This will finally result in a situation where most of the routine support for the business process is automated and some identified exceptions can be handled by the software agents. Within the simulation, some of the identified behaviours are modelled in the simulated humans, and some are modelled in the simulated (actually emulated) software agents. Because both these agent types are in fact software agents, it is possible to transfer behaviour descriptions from simulated humans to the emulated software agents. This transfer represents the incremental development via process gaming of the multi-software-agent system. By transferring behaviour, a more subtle transformation takes place, that is, the humans are empowering the software agents, by delegating to them those tasks that are considered routine and repetitive and should be automated. During the simulation sessions, the players and the experimenters should be able to see and understand the simulated execution of the business process. This is achieved via a TALL editor that dynamically visualizes the progress of the business process using the TALL modelling constructs. During escape sessions, the players and the experimenter (i.e. the MAS developer) are able to operate modifications in the structure (agent, role, interaction) and behaviour of the agents via an editor tool existent in AGE.

ESCAPE MODE / INTERVENTION IN SIMULATION

The exceptions that trigger the escape mode mentioned in the previous sections occur when agents encounter inconsistencies in their models. In the following, we give a short example of a game in AGE, illustrated with TALL interaction behaviour diagrams. This takes places in a logistic environment, and considers the existence of three agents: a sales manager, a tactical production planner, and a shopfloor level scheduler. The sales manager is in charge with the relation with the customer, and one of his main responsibilities is to expedite orders if customers ask for it (for a price). The planner is in charge with the internal movement of goods through the entire business, from purchasing to sales, and his main role is to keep inventories and flows in balance. To insure this, he has at his disposal a set of powerful IT tools for planning but also execution scheduling. The scheduler is the person responsible for the shopfloor level execution, and usually he is not using IT tools. If he needs a schedule, the best way is to ask the planner to generate one for him. This arrangement is typical in production organisations where the shopfloor has a so called “jumble” architecture, that is, the flow of materials and manufacturing tasks is difficult to organise in the line-style, or dock-style. Moreover, changes in the shopfloor schedules have to be registered immediately at tactical level. In this way scheduling remains an activity that is still linked to the planner and not the scheduler, who is in charge with execution time ad-hoc changes.

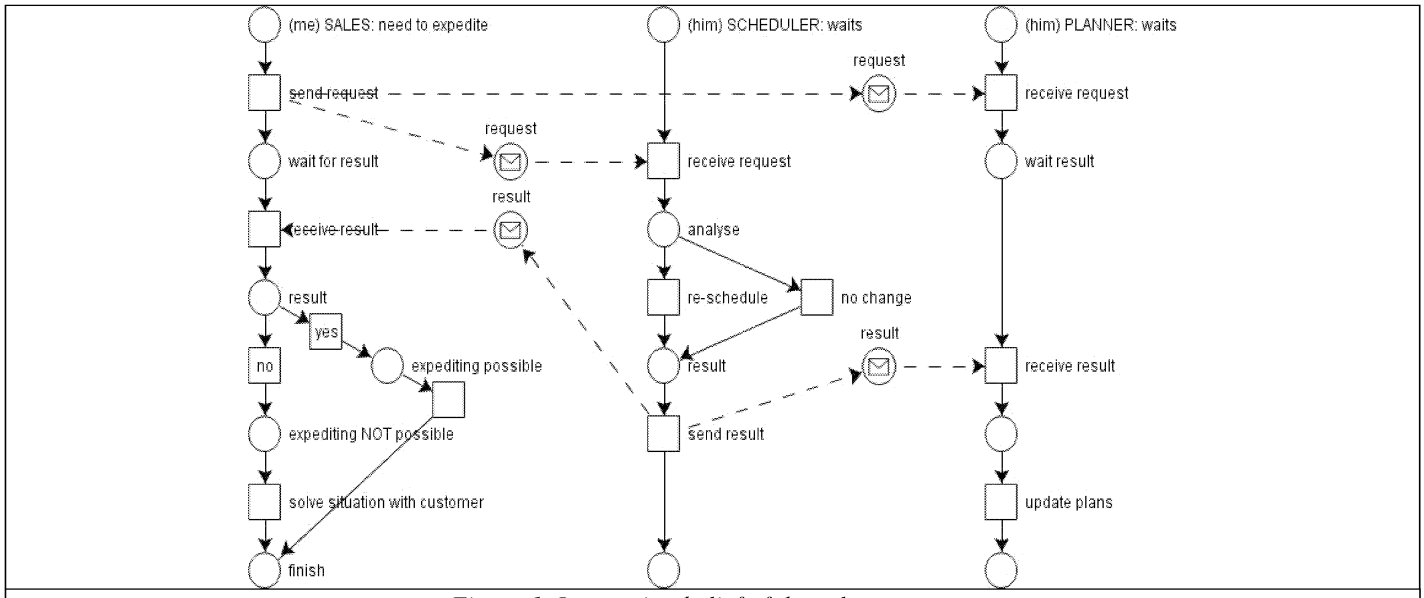


Figure 1. Interaction belief of the sales manager.

The figures illustrate the situation of *order expedition*. The behaviours of the agents are depicted as interaction belief diagrams. First, in figure 1, we show the interaction belief of the sales manager. He intends to send a request to the scheduler, and a note to the planner (who is finally responsible for the actions to execute the order faster). After, he waits for the response, and finishes its actions depending on the result. According to the expected behaviours depicted in the other swimlanes, the scheduler is expected to analyze the request and make himself a re-schedule if possible. Then, it sends a message back to the sales manager with the positive result. The planner is expected by the sales manager to wait for an answer from the scheduler, after it receives the request for re-scheduling. In this case, the sales manager is not completely aware of the way things are working in the organisation. He believes that the scheduler is doing the scheduling, but in reality, the planner is in charge with that. In terms of syntax, the process description for the 're-scheduling interaction' has to contain three swimlanes. One describes the intended activities to be performed by the sales manager (left). The other swimlanes describe what the sales manager's beliefs about the scheduler (middle) and the planner are (right). Swimlanes identify not only activity areas, but also the identity of the participants in the interaction. We label the swimlane with a 'role-name'. The first left-side swimlane is the one illustrating the 'me' behavior. We opted for adding supplementary information to the role name that identifies a swimlane, in the form of a predicate *me*, with the attributes agent id and agent type. Albeit being somehow redundant, that allows the analyst to see immediately who the owner of this behavior description is.

In figure 2, the interaction belief of the scheduler is depicted, by using also three swimlanes. In this organisation, the scheduler is "far" from the sales manager. Normally, he is never directly working with him. However, due to personnel hierarchy reasons, he is not refusing directly a request from a sales manager, but he is answering that the re-scheduling is not possible. In any situation, a request will end with a failure to expedite. The behaviour of the scheduler is tuned to the fact that the planner is the one who is able to produce quickly and accurate tentative schedules for the shopfloor. If the request for expediting would have come directly from the planner, accompanied by a proposed schedule, the chances for a positive response would be much higher.

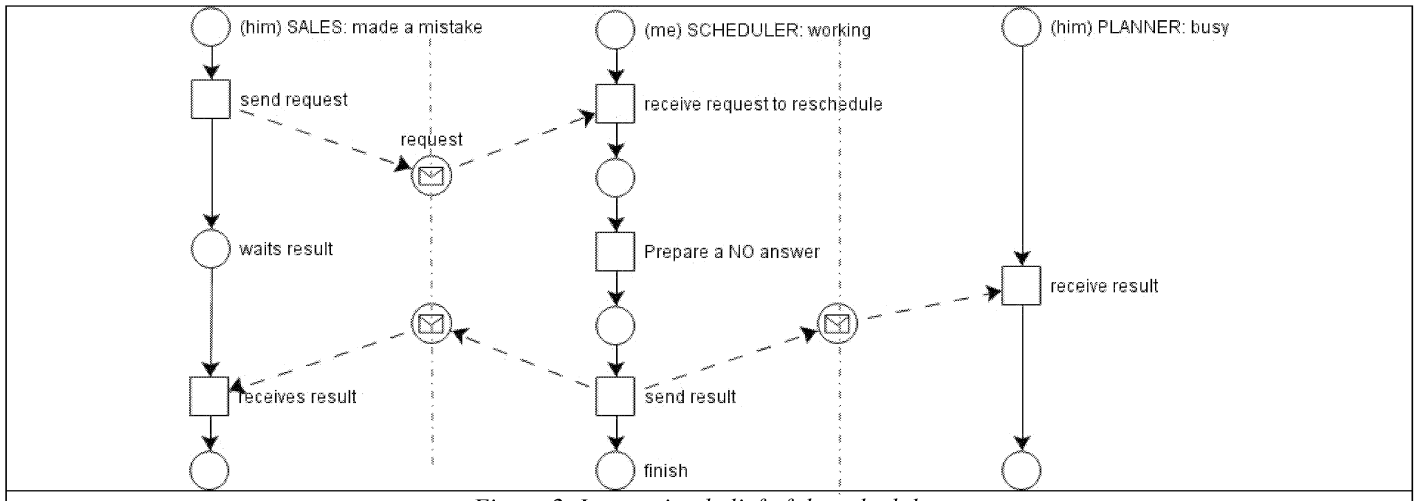
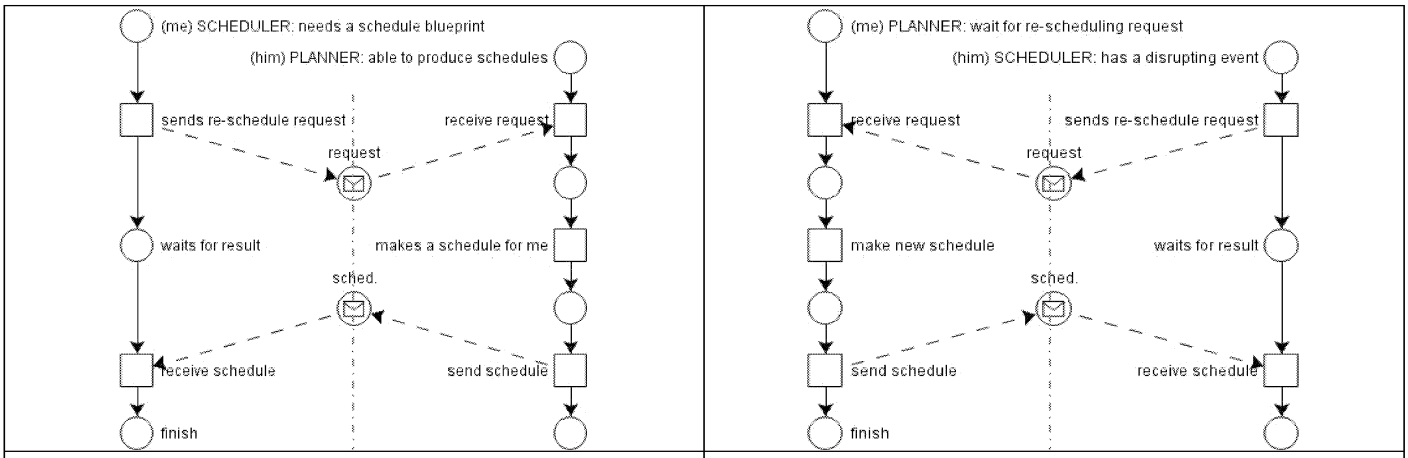


Figure 2. Interaction belief of the scheduler.



(a) IB of scheduler *(b) IB of planner*

Figure 3. The agents' interaction beliefs.

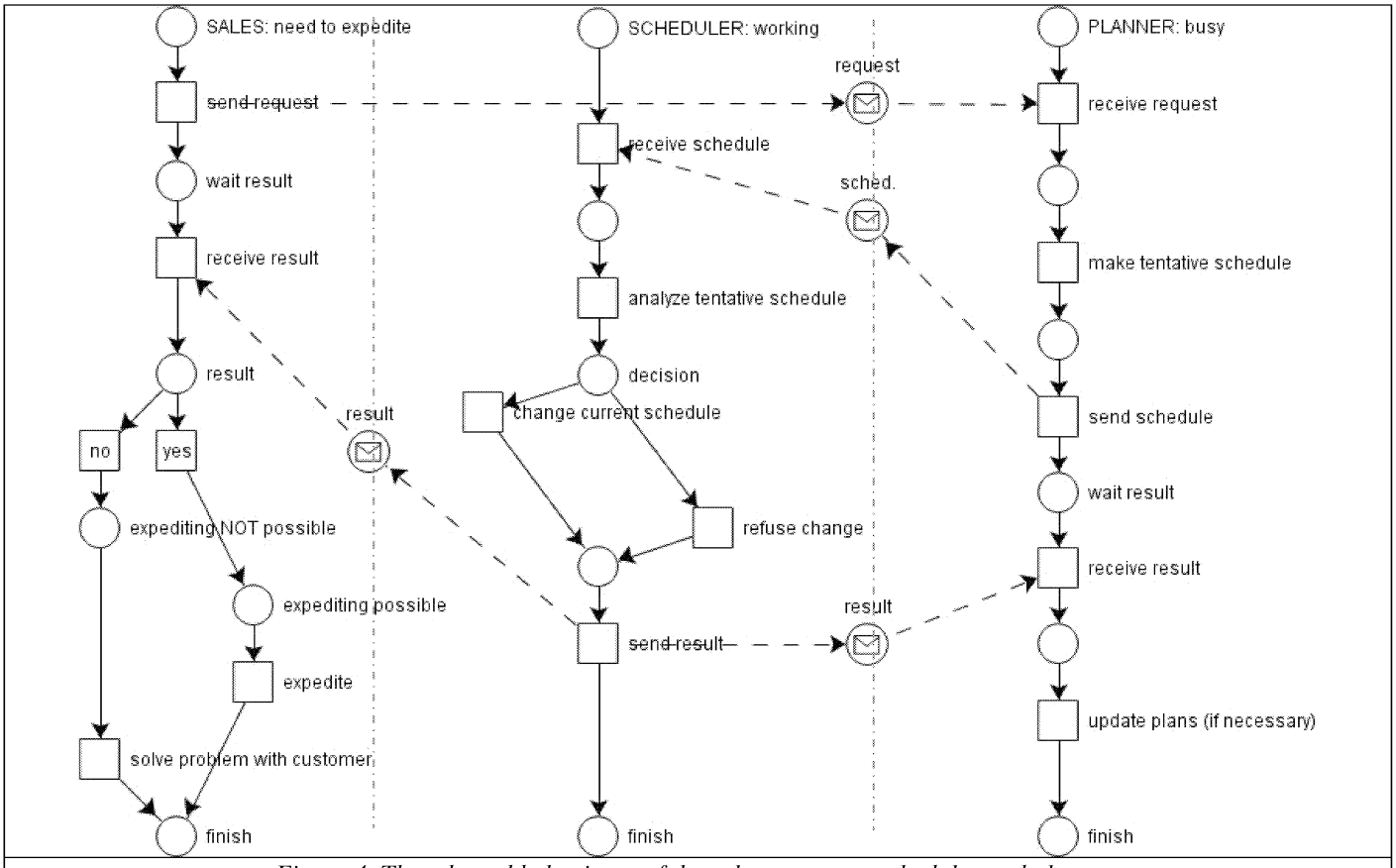


Figure 4. The adapted behaviours of the sales manager, scheduler and planner.

In the sense of pure communication channels, the expected messages and their sequence lead to matching behaviours between the sales manager and the scheduler. They both send the correct messages at the right moment. However, the expectation that the scheduler is willing to re-schedule is a false assumption on the part of the sales manager. During simulation execution, because the agents are not blocked, their behaviour is performed without fail and does not trigger automatically the escape mode. That means that the repetitive negative response of the scheduler can only be discovered by an external observer of the process. In case that this observer exists as the game player and he is able to find a solution (for example by changing the behaviour of the sales manager), and such an action qualifies as an intervention. A kind of trigger for escape could be a simple rule for the sales manager that after a certain number of negative responses, he automatically invokes the intervention. In a more generic setting, low performance indicators of various agents can be used as escape mode triggers implemented as self-reflective rules. Considering the behaviour of the scheduler once more, we have to emphasise that he *is* aware that the planner is able to generate plans. Figure 3 depicts two interaction beliefs from the perspective of the scheduler (a) and the planner (b) of how this interaction takes place. When the scheduler has a problem, and needs a new schedule, he asks the planner to provide him with one generated by its automatic capability.

If these three agents would exchange their beliefs about their behaviours, the sales manager would realize the response from the scheduler is always negative, and that the planner is actually responsible for the re-scheduling. In the case that the player intervenes after the escape of one of the sales managers, due to poor performance for example, the corrected situation, in terms of a proposed workflow between the three agents, could be the solution depicted in figure 4. Here, the request is always sent first to the planner, who makes a tentative schedule, which is sent to the shopfloor, where it could be carried out or not. The agents, via the players, can learn this new behaviour and for the time being, the interaction behaviours will be aligned. If we consider that a new sales manager appears in the environment, and he does not know that he has to expedite orders through the planner and not the scheduler, that will mean that we are back to the situation presented in figure 1. To avoid this situation, the simplest solution is to enable the behaviour of the scheduler with an escape trigger. In figure 2, the scheduler was allowed to receive a request message from the sales manager. This should remain always possible, in order to keep the communication channels open. However, due to the fact that there is no longer a possibility modelled in the behaviour of the scheduler (which changed from the one in figure 1 to the one in figure 4), receiving the message will automatically trigger an escape, because the scheduler agent artificially "does not know" what to do with a message from the sales manager.

CONCLUSIONS

In real organisations, as the one discussed in the above example, this pattern of error detection, correction from hierarchically superior participants, and local learning by agents, is one of the most basic ways to adjust to various forms of change (Horling & Lesser, 2005). Change can appear as an infusion of new ideas, appearance /disappearance of new agents in the organisation, pressure from the environment to change, etc. This pattern of adapting to change, is typical for trial and error based learning in interactional processes and it is natural to use methods inspired from business gaming. That means that the software agents change their behaviour according with the new understandings of their owners/players, who respond to escapes triggered by built-in mechanisms of the software agents. They may also intervene when they observe that the behaviour of their agents is not aligned with the behaviour of other agents. Escape and intervention may occur in the simulation phase of the process, in a controlled environment, but it can also happen during the usage of a real multi-agent system that supports a running business process. By enriching incrementally the behavior of the agents, we can say that agents "grow" (Ferber & al, 2004). The escape/intervention rate decreases and the simulation does not have to be controlled from "outside" by the human players and experimenters. With this growth, control of the overall process will emerge "inside-out". Our main message in the full paper will be that the growth of the agents during interactive simulation (e.g. gaming) (and after), represents in a sense, another form of organisational learning, where implicit distributed knowledge of the participants in a business process is made explicit by capturing it in terms of software agent coding. These agents can be used for personnel training, simulation and gaming leading to business process re-engineering, and finally, development of MAS that support the humans during business process execution. Participative simulations can be used after the MAS deployment as well, exploring new ways to support the process, and enriching the software agents' behaviour. Looking at the general relevance of this approach, it is actually about the increase of the knowledge worker productivity in a process-view dominated environment. The last years saw a diminished autonomy (Riss et al, 2005) of the knowledge worker in managing themselves and applying continuous innovation, despite the real need for this autonomy.

REFERENCES

Berry, B.J.L., 2002. Adaptive agents, intelligence, and emergent human organization: Capturing complexity through agent-based modelling, *PNAS*, vol. 99, 7187-7188.

Bonabeau, E., 2002. Agent-based modeling: Methods and techniques for simulating human systems, *PNAS*, vol. 99, 7280-7287.

Carley, K., 2000. *Computational Analysis of Social and Organizational Systems.* Organizational Science, 34(2).

Daft, R.L., 2004. *Organization Theory and Design.* 8th Ed. South-Western College Publishing, Cincinnati.

Ferber. J., Gutknecht, O., & Michel, F., 2004. From Agents to Organizations: an Organizational View of Multi-Agent Systems. *LNCS*, 2935, 214-230.

Fisher, D. M., 2004. The Business Process Maturity Model, A Practical Approach for Identifying Opportunities for Optimization, *Business Process Trends,* Sept.-04, 1-7.

Horling, B., and Lesser, V., 2005. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 19 (4), 281-316.

Lainema, T., 2004. *Redesigning the Traditional Business Gaming Process – Aiming to Capture Business Process Authenticity*, Journal of Information Technology Education, 4/04.

Meyer, G. G., Szirbik, N., 2007. Anticipatory Alignment Mechanisms for Behavioural Learning in Multi Agent Systems, *Lecture Notes in Artificial Intelligence, (eds. M. V. Butz et al.),* vol. LNAI 4520, 325–344, Springer-Verlag.

Morgan, G., 2006. *Images of Organization*. Updated ed. Sage Publications, Thousand Oaks.

Nonaka I, Takeuchi H., 1995.*The knowledge-creating company*. Oxford, UK: University Press.

Polyani, M., 1966. *The Tacit Dimension*. Rutlege & Keegan, London.

Roest, G. G., Szirbik, N., 2007. An Organisational Pattern Inspired from Interactional Processes, *Proc. of The 6th International Workshop on Social Intelligence Design, (eds. A. Nijholt, O. Stock, T. Nishida),* 141-155, Trento, 2-4 July, 2007, published by CTIT, University of Twente.

Riss, U. V., Rickayzen, A., Maus, H., Aalst, van der, W. M. P., 2005. Challenges for Business Process and Task Management, *Journal of Universal Knowledge Management,* no. 2, 77-100.

Stuit, M., Szirbik, N., 2007. Modelling and Executing Complex and Dynamic Business Processes by Reification of Agent Interactions, *Lecture Notes in Artificial Intelligence, (eds. G. O'Hare et al.),* vol. LNAI 4457, 106-125, Springer-Verlag.

Wagner, G., 2003. The Agent-Object-Relationship metamodel: towards a unified view of state and behaviour. *Information Systems*, 28, 475-504.

Wooldridge, M., 2002. *Introduction to Multi-Agent Systems*, John Wiley and sons, Chichester, England.

SOCIAL AGENTS

MASCEM IMPROVEMENT FOR AGENT BASED COALITIONS

Isabel Praça, André Andrade, Hugo Morais, Carlos Ramos, Zita Vale
GECAD –Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão
Instituto Superior de Engenharia / Instituto Politécnico do Porto
Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal
Fax: +351-228321159, Phone: +351-228340500
E-mail: icp@isep.ipp.pt

KEYWORDS
Simulation, Decision-making, Intelligent Agents, Coalitions, Electricity Markets.

ABSTRACT

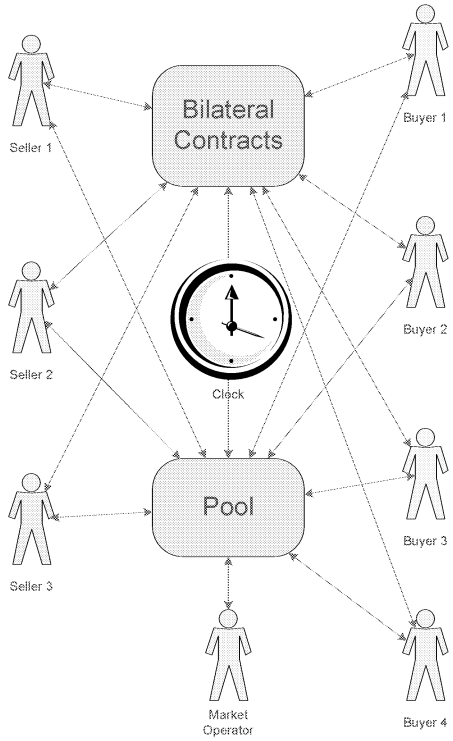
This paper presents MASCEM - Multi-Agent Simulator for Electricity Markets improvement towards an enlarged model for Seller Agents coalitions. The simulator has been improved, both regarding its user interface and internal structure. The OOA, used as development platform, version was updated and the multi-agent model was adjusted for implementing and testing several negotiations regarding Seller agents' coalitions. Seller coalitions are a very important subject regarding the increased relevance of Distributed Generation under liberalised electricity markets.

INTRODUCTION

The development of new low emission generation tech-nologies (wind generation, solar cells, fuel cells, micro-turbines) leads us to rethink the location of a significant part of the production: distributed generators owned by decentralized players will provide a significant amount of the electricity generation.
All over the world Distributed Generation is seen as a valuable help to get cleaner and more efficient electricity. With Distributed Generation, electricity is produced near consumers' location and not transmitted over long distances. Thus, it is possible to get lower losses. Moreover, new generation technologies, mainly based on renewable resources, with environmental advantages are a key issue for sustainable development. Investments in this field are encouraged by a favourable regulatory framework and the equipment costs are more attractive every day. At this stage, the main technologies used are micro and hydro turbines, fuel cells, wind generation, and solar cells.
A deregulated market where every single low power rating generation unit sells its power on the market would be optimal for the whole community both economically and technically.
To get negotiation power and advantages of scale econ-omy, these players can be aggregated giving place to a new concept: the Virtual Power Producer (VPP). VPPs are multi-technology and multi-site heterogeneous entities. VPPs should adopt organization and management meth-odologies so they can make Distributed Generation a really profitable activity.
In order to operate in an efficient way, VPPs should have adequate decision-support tools. These must be based on the availability and processing of the required information and knowledge concerning producers and market operation. A successful achievement of VPPs' goals requires the use of a mix of adequate technologies for optimizing and supporting their activities.
Agents and multi-agent systems that adequately simulate electricity markets behaviour are essential tools to gather knowledge to provide decision-support to strategic behaviour.
MASCEM– Multi-Agent Simulator of Competitive Electricity Markets (Praça et al. 2003) was developed to study several negotiation mechanisms usually found in electricity day-ahead markets. In MASCEM market participants have strategic behavior and a scenario decision algorithm to support their decisions. With MASCEM several experi-ences have already been made, leading us to achieve some conclusions and define future developments. One of the most important goals of MASCEM is the simulation of several different electricity market mechanisms.
With VPP integration into MASCEM, VPP proposals can be effectively evaluated under scenarios considering several different market mechanisms and containing several other producers and consumer entities. With MASCEM, VPP can improve their strategies to face mar-ket.

MASCEM OVERVIEW

Our Simulator facilitates agent meeting and matching, besides supporting the negotiation model. In order to have results and feedback to improve the negotiation models and consequently the behaviour of user agents, we simulate a series of negotiation periods, $D = \{1,2,...,n\}$, where each one is composed by a fixed interval of time $T = \{0,1,...,m\}$. At a particular negotiation period, each agent has an objective that specifies its intention to buy or sell and on what conditions.
The available agents can establish their own objectives and decision rules. Moreover, they can adapt their strategies as the simulation progresses on the basis of previous effort's successes or failures. The simulator probes the conditions and the effects of market rules, by simulating the participant's strategic behaviour. Figure 1 illustrates the Multi-Agent Model.



Figures 1: MASCEM negotiation framework

One of the main objectives of electricity markets is to decrease electricity costs through competition. Several market structure models exist that could help achieve this goal. The market environment typically consists of a Pool, as well as a floor for Bilateral Contracts.

A Pool is a marketplace where electricity-generating companies submit production bids and their corresponding market prices, and consumer companies submit consumption bids. A Market Operator regulates the pool. The Market Operator uses a market-clearing tool to set market price and a set of accepted production and consumption bids for every hour. In Pools, an appropriate market-clearing tool is an auction mechanism.

Bilateral Contracts are negotiable agreements between two traders about power delivery and receipt. The Bilateral-Contract model is flexible; negotiating parties can specify their own contract terms.

The Hybrid model combines features of Pools and Bi-lateral Contracts (Praça et al. 2005). In this model, a Pool isn't mandatory, and customers can either negotiate a power supply agreement directly with suppliers or accept power at the established market price. This model therefore offers customer choice.

There are several entities involved in the negotiations; we propose a multi-agent model to represent all the involved entities and their relationships.

MASCEM multi-agent model includes: a Market Facilitator Agent, Seller Agents, Buyer Agents, Trader Agents, a Market Operator Agent and a System Operator Agent. Three types of markets are simulated: Pool Markets, Bilateral Contracts and Hybrid Markets.

The Market Facilitator is the coordinator of the market. It knows the identities of all the agents present in the market, regulates the negotiation process and assures the market is functioning according to the established rules. The first step agents' have to do is the registration at the Market Facilitator, specifying their market role and services.

Seller and Buyer Agents are the two key players in the market. Sellers represent entities able to sell electricity in the market, e.g. companies holding electricity production units. Buyers may represent electricity consumers or even distribution companies. The user, who must also specify their intrinsic and strategic characteristics, defines the number of Sellers and Buyers in each scenario. By intrinsic characteristics we mean the individual knowledge related to reservation and preferred prices, and also to the available capacity (or power needs if it is a Buyer). By strategic characteristics we mean the strategies the agent will employ to reach its objectives.

Sellers will compete with each other, since they are all interested in selling all their available capacity and in obtaining the highest possible market quote. On the other hand, Sellers will cooperate with Buyers while trying to establish some agreement that is profitable for both. This is a very rich domain where it is possible to develop and test several algorithms and negotiation mechanisms for both cooperation and competition.

The System Operator Agent represents the responsible for the transmission grid and all the involved technical constraints. Every contract established, either through Bilateral Contracts or through the Pool, must first be communicated to the System Operator, who analyses its technical viability from the Power System point of view (e.g. feasibility of Power Flow to attend all needs).

The Market Operator Agent represents the responsible for the Pool mechanism. This agent is only present in simulations of Pool or Hybrid markets. The Market Operator will receive bids from Sellers, Buyers and Traders, analyse them and establish the marginal price and accepted bids. The process of determining the accepted bids is done according to the technical validation by the Sys-tem Operator, after, the Market Operator communicates to Sellers, Buyers and Traders the acceptance, or not, of their bids and, optionally, the market price.

Agents Strategic Behaviour

On the basis of the results obtained in a negotiation period Sellers, Buyers and Traders revise their strategies for the next period. Seller, Buyer and Trader Agents have strategic behavior to define their desired price. These agents have time-dependent strategies, to change the price according to the remaining time until the end of the negotiation period; and behavior-dependent strategies, to define the next period price according to the results obtained in the previous ones.

MASCEM implements four types of strategies to change the price during a negotiation period: Determined, Anxious, Moderate and Gluttonous. The difference between these strategies is the time instant at which the agent starts to modify the price and the amount it changes. Although time-dependent strategies are simple to understand and implement (Morris et al. 2003), they are very important since they allow the simulation of important issues such as: emotional aspects and different risk behaviors. For example: an agent using a

Determined Strategy is a risk indifferent one; while Gluttonous agents exhibit the behavior more risk disposable, since they maintain the same price until very close to the end of the negotiation period, taking the risk of not selling. To adjust price between negotiation periods, also referred as behavior-dependent strategies, two different strategies were implemented: one called Composed Goal Directed and another called Adapted Derivative Following, see details in (Praça et al. 2005). These are important strategies that use the knowledge obtained with past experiences to de-fine bid prices for next periods.

To obtain an efficient decision support, Seller and Buyer agents also have the capability of using the Scenario Analysis Algorithm.

This algorithm provides a more complex support to develop and implement dynamic pricing strategies since each agent analyses and develops a strategic bid, for the next period, taking into account not only their previous results but also other players results and expected future reactions.

SIMULATOR IMPROVEMENT

MASCEM was initially developed on the top of the Open Agent Architecture (OAA) version 2.0.12, while agents were developed using Java Virtual Machine 1.2.2.

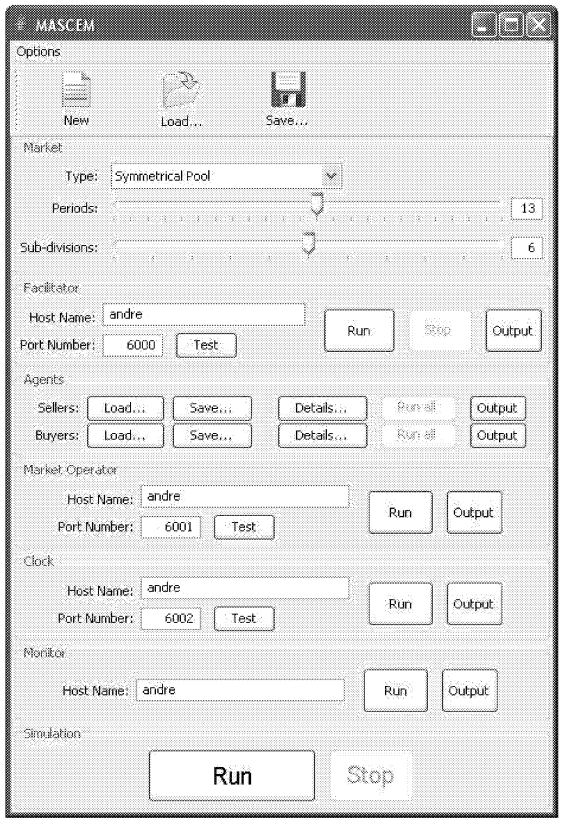
OAA (http://www.ai.sri.com/~oaa/) is a framework for integrating a community of heterogeneous software agents in a distributed environment. It is structured to minimize the effort in creating new agents, written in various languages and operating platforms; to encourage the reuse of existing agents; and to allow the creation of dynamic and flexible agent communities.

The OAA's Interagent Communication Language is the interface and communication language shared by all agents, no matter which machine they are running on or which programming language they are programmed in. OAA is not a framework specifically devoted to develop simulations; some extensions were made to make it more suitable, such as the inclusion of a clock to introduce the time evolution mechanism of the simulation.

In the meanwhile, several updates were done to OAA infrastructure, whereas its version is now 2.3.1. MASCEM was then updated and now is implemented on the top of OAA version 2.3.1, using OAA AgentLib library, and Java Virtual Machine 1.5.0. This implied several work concerning code structure.

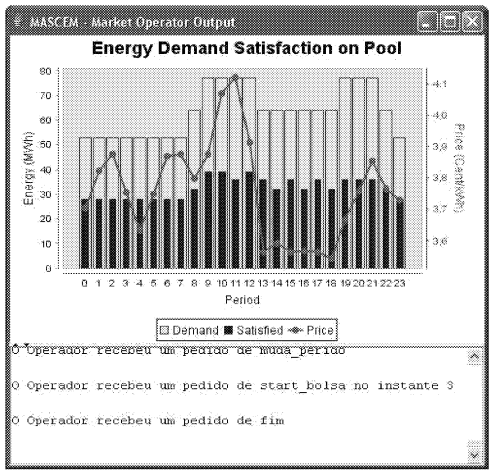
At the same time, MASCEM interface was improved, to have flexible and user-friendly interface. This is a very important issue regarding the usefulness of simulators, and particularly MASCEM usefulness.

Interface screens for setting agents characteristics, regarding preferences and strategies to use, were also improved. Concerning model distribution over a network, it is possible to set the machine and related port where each agent will be run from. Figure 2 illustrates MASCEM main interface.



Figures 2: MASCEM main screen

Another important issue is the improvement of the interface regarding simulation evolution and results analysis. Both market and agents' results may be observed by means of Stacked Column charts. Figure 3 shows interface for Pool market results.



Figures 3: MASCEM interface for Pool market results

It is important to note that concerning a detailed simulation analysis, it is possible to see messages exchanged between agents during the negotiation process. At the bottom of the screen shown on Figure 3 we can see messages regarding Pool negotiation evolution process.

MULTI-AGENT MODEL IMPROVEMENT

The aggregation of Distributed Generation plants gives place to the new concept of Virtual Power Producers (VPP). VPPs are multi-technology and multi-site heterogeneous entities. Any type of generation unit or load may be included: wind turbines, photovoltaic, mini turbines, micro-turbine, fuel cells, energy storage units, non-controllable loads, controllable loads etc. The typical size of single distributed energy resource units may range from a few kW to some MW.

In the scope of a VPP, aggregated producers (AP) can make sure their generators are optimally operated and that the power that is not consumed in their installation has good chances to be sold on the market. At the same time, VPPs will be able to commit to a more robust generation profile, raising the value of non-dispatchable generation technologies.

Under this context, VPPs can ensure secure, environmentally friendly generation and optimal management of heat, electricity and cold and optimal operation and maintenance of electrical equipment, including the sale of electricity to the EM. VPPs should adopt organization and management methodologies so that they can make Distributed Generation a really profitable activity.

VPPs must be flexible enough to use the advantages of its resources (e.g. market-based environmental value in the form of pollution and/or carbon credits, renewable energy credits) while overcoming their problems and limitations.

VPPs must identify the characteristics of each of the AP and try to optimize the selling activity so that each associate delivers the biggest possible amount of energy. However, this is not simple due to generation uncertainty concerning technologies that are dependent from natural resources such as wind, sun, waves or water flows.

So, in order to have VPP able to coexist with other market agents, it is necessary to have profits and to have credibility in the EM. This must be considered in VPPs organization and operation methodologies as their goal is to optimize their APs' profits in the market.

A successful achievement of VPPs' goals requires the use of a mix of adequate technologies for optimizing and supporting their activities. Under this scope agents and multi-agent systems are important technologies to adequately simulate EM behavior and gather knowledge to provide decision-support to strategic behavior.

Taking into account the already described MASCEM characteristics, it can be a valuable framework to test VPP functioning under different market mechanisms and concerning different market strategic behavior.

Coalitions in Multi-Agent Systems

Coalition formation is the coming together of a number of distinct, autonomous agents that agree to coordinate and cooperate, acting as a coherent grouping, in the performance of a specific task. Such coalitions can improve the performance of the individual agents and/or the system as a whole. It is an important form of interaction in multi-agent systems.

It has been advocated in e-commerce (where buyers may pool their requirements in order to obtain bigger group discounts), in grid computing (where multi-institution virtual organizations are viewed as being central to coordinated resource sharing and problem solving), and in e-business (where agile groupings of agents need to be formed in order to satisfy particular market niches). In all of these cases, the formation of coalitions aims to increase the agents' abilities to satisfy goals and to maximize their individual or the system's outcomes.

Most work on coalition formation in Multi-Agent systems and game theory has focus on payoff distribution, where it is usually assumed that a coalition structure has been formed, and the question is then how to divide the payoff so that the coalition structure is stable. In this context, many solutions have been proposed based on different stability concepts. Transfer schemes have also been developed to transfer non-stable payoff distributions to stable ones (while keeping the coalition structure unchanged).

There are several works concerning essentially the coalition structure generation (Sandholm et al. 1999) (Dang and Jennings 2004). The work of Shehory and Kraus (Shehory and Kraus 1996) considers a somewhat broader environment, where the coalitions can be overlapped but the complexity is reduced by limiting the size of the coalitions.

Some other researchers address both coalition structure generation and payoff distribution in competitive environments. Ketchpel (Ketchpel 1994) presents a coalition formation method with cubic running time in the number of agents, but his method can neither guarantee a bound from the optimal nor stability. Shehory and Kraus's protocol guarantees that if the agents follow it, certain stability (kernel-stability) is met. In the same paper, they also present an alternative protocol that offers a weaker form of stability with polynomial running time. However, in both cases, no bound from the optimal is guaranteed.

More recent research in coalition formation area has also begun to pay attention to dynamic environments, where agents may enter or leave the coalition formation process and many uncertainties are present (e.g. the coalition value is not fixed, but it is context-based (Klush and Gerber 2002).

VPP Coalition establishment in MASCEM

A negotiation mechanism regarding coalition formation under the scope of a VPP is in fact being included in MASCEM, and strategies will be developed considering the three phases of a coalition's formation process.

VPP needs to have an adequate knowledge of each potential aggregated producer characteristics. We are working on a multi-criteria negotiation protocol.

Some of the most important characteristics are:

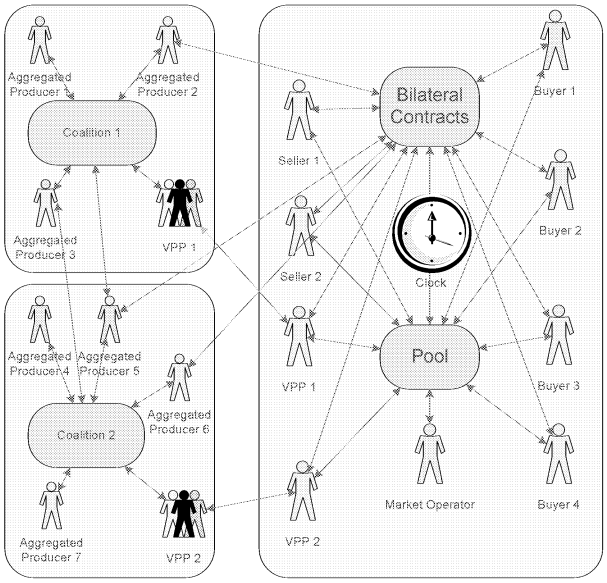
- Nominal Power: the sum of nominal power installed in each producer;
- Available Power: the power a VPP can buy to the producer;
- Overload Power: some units may produce overload power for limited periods. The VPP may use this power in critical situations;
- Equipment characteristics: information concerning producers' equipment allows the VPP to know the

power characteristic, reliability, maintenance periods, lifetime, relation with external factors, possible variations of the energy price in function of the cost of the primary resources, etc.;

- Operating limits: for the units which are dependent on natural resources, it is possible that the resource is below or above equipment operating limits. This must be considered in risk analysis of the generation forecast. Usually when the resources forecast is near to the minimum machines operating limit the risk is small, but when they are near to the maximum limit the risk can be enormous;
- Grid connection characteristics: This is an important aspect if it is necessary to pay the losses in the lines; also the existence of two or more producers connected to the same electric substation should be considered; etc;
- Historical generation data: the availability of historic generation data can enable the VPP to get useful forecasting tools.

These issues must be carefully analyzed before a new producer enters the coalition. A multi-criteria decision function is being developed. This function must be a dynamic one, since the characteristics weights depends on the lack of already aggregated producers.

Regarding VPP market participation, essentially the same market interface as Seller or Buyer agents is used. Figure 4 illustrates MASCEM negotiation framework regarding VPPs.



Figures 4: MASCEM negotiation framework regarding VPPs

However, there are some preliminary steps to define its proposals and to divide market results among VPP members. First all the capacity available from the different aggregated producers must be gathered, to establish the electricity amount to trade on the market, and the different production costs analyzed to define the interval for acceptable proposals. This means VPP agents will have a utility function that aggregates all the involved units' characteristics. The analysis of received proposals, and payoff distribution, will be done according to each unit capabilities and costs.

Important is the use of previsions, such as climatic ones, to update the knowledge of aggregated producers and have better forecasts regarding the energy they will be able to provide to the VPP.

CONCLUSIONS

In this paper we present MASCEM simulator improvements. MASCEM was updated, both concerning technical issues, such as platform version update, and market issues, such as the increase of Distributed Generation, leading to the emergence of new entities, the Virtual Power Producer, which represents a coalition of small producers.

MASCEM is now able to simulate scenarios concerning VPPs occurrence on electricity markets. These scenarios may be analysed from the producers' point of view, concerning income increase by entering a VPP coalition; from the coalition point view, concerning both coalition structure generation and payoff distribution in competitive electricity markets; and from the market point of view, concerning market behaviour adjustments in the presence of VPP. Currently we are working on the multi-criteria decision function for dynamic coalition generation and evolution, and protocols for inside coalition negotiations.

ACKNOWLEDGEMENTS

The authors would like to acknowledge FCT, FEDER, POCTI, POSI, POCI and POSC for their support to R&D Projects and GECAD Unit.

REFERENCES

Dang, V. and N. Jennings. 2004. "Generating coalition structures with finite bound from the optimal guarantees". In *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems*. (Nova York, E.U.A). 564-571.

Ketchpel, S. P. 1996. "Forming coalitions in the face of uncertain rewards". *In Proceedings of the Twelfth National Conference on Artificial Intelligence*. 414–419.

Klusch, M. and A. Gerber. 2002. "Dynamic coalition formation among rational agents". *IEEE Intelligent Systems* 17, Nº 3, 42–47.

Morris, J., A. Greenwald, and P. Maes. 2003. "Learning Curve: A Simulation-based Approach to Dynamic Pricing". *Electronic Commerce Research: Special Issue on Aspects of Internet Agent-based E-Business Systems* 3, Nº 3-4. Kluwer Academic Publishers, 245-276.

Praça, I., C. Ramos, Z. Vale, and M. Cordeiro. 2003. "MASCEM: A Multiagent System that Simulates Competitive Electricity Markets". *IEEE Intelligent Systems* 18, Nº 6, 54-60.

Praça, I., C. Ramos , Z. Vale, and M. Cordeiro. 2005. "Intelligent Agents for Negotiation and Game-based Decision Support in Electricity Markets". *International Journal of Engineering Intelligent Systems, Special Issue "Intelligent Systems Application to Power Systems"* 13, Nº 2, 147-154.

Sandholm, T., K. Larson, M. Andersson, O. Shehory and F.Tohme. 1999. "Coalition structure generation with worst case guarantees". *Artificial Intelligence* 111, Nº 1-2, 209– 238.

Shehory, O. and S. Kraus. 1996. "A kernel-oriented model for coalition-formation in general environments: Implementation and results". In *Proceedings of the "Thirteenth National Conference on Artificial Intelligence"*, 134–140.

MODELLING GROUP DECISION SIMULATION THROUGH ARGUMENTATION

Goreti Marreiros[1], Paulo Novais[2],José Machado[2], Carlos Ramos[1] and José Neves[2]

[1] GECAD – Knowledge Engineering and Decision Support Group
Institute of Engineering – Polytechnic of Oporto
{goreti, csr}@dei.isep.ipp.pt

[2] Department of Informatics - CCTC
University of Minho
{pjon, jmac, jneves}@di.uminho.pt

KEYWORDS
Argumentation, Group Decision Making, Agents Based Simulation.

ABSTRACT

Group decision making plays an important role in today's organisations. The impact of decision making is so high and complex, that rarely the decision making process is made individually. In Group Decision Argumentation, there is a set of participants, with different profiles and expertise levels, that exchange ideas or engage in a process of argumentation and counter-argumentation, negotiate, cooperate, collaborate or even discuss techniques and/or methodologies for problem solving. In this paper, it is proposed a Multi-Agent simulator for the behaviour representation of group members in a decision making process. Agents behave depending on rational and emotional intelligence and use persuasive argumentation to convince and make alternative choices.

INTRODUCTION

Group decision and negotiation terms find themselves misleading. However, group decision and negotiation processes are quite different. In group decision making, the alternatives are already settled (in the choice moment); in negotiation the alternatives are sequentially presented and modified for the involved parties. The information sharing process is usually more important in group decision. The voting mechanisms can be a possibility; in negotiation they do not make sense. In a negotiation process, it is more frequent that one of the parties abandons the negotiation. The main objectives in group decision are usually shared by all the parties, while in negotiation they are usually antagonist. However, in some contexts, the group members may have quite different goals, that will imply the use of negotiation in order to achieve an acceptable outcome.

Group decision making and negotiation processes represent very complex human activities. A better understanding of those processes implies the relation of several disciplines, such as psychology, sociology, political science, etc (Turban and Aronson 1998).

Along the last 20 years, several GDSS were developed, some dedicated to be used exclusively in decision rooms and others with features to support ubiquitous group decision meetings (Karacapilidis and Papadia 2001; Group Systems/URL).

Recent growing interest in artificial agents and their potential application in simulation areas such as: individual decision making ("what-if" scenarios), e-commerce (to simulate the buyers and sellers behaviour), crisis situations (e.g. fire combat simulation), traffic simulation, military training, entertainment (e.g. movies) have given in-creased importance to automated negotiation (Ito and Shintani 1997; Kudenko et al. 2003; Zamfirescu et al. 2001; Payne et al. 2000).

In the multi-agent literature, various interaction and decision mechanisms for auto-mated negotiation have been proposed and studied. Traditional mechanisms include game-theoretic analysis and heuristic-based approaches, other approach to automated negotiation are the argumentation-based approaches (Amgoud et al. 2005; Jennings et al. 1998; Novais at al. 2005).

In the context of negotiation, argumentation is viewed as a mechanism to make possible the information exchange. An argument is viewed as a piece of information that may allow for an agent to (Sierra et al. 1998):

- Justify its negotiation decision or option; and
- Influence others agents about the quality of its proposals.

A group decision making process involves multiple actors, each one with different expertise, preferences, perspectives of the problem and different emotional states. Many times, different types of conflicts and disagreements arise, and it is necessary to overcome them. Argumentation is used in every day dialogues by parents, students, doctors to justify choice or to convince the interlocutor about their point of views. Also, in group decision domain, argumentation can be an excellent way to justify possible choices and to convince other elements of the group about the best or worst alternative.

It is proposed a multi-agent model to simulate group decision making processes, where agents are designed with emotional properties and reason with incomplete in-formation. Agents use persuasive argumentation for convincement and the best choice. The application area of this model is the multi-criteria decision problems. The multi-criteria decision process aims to evaluate a set of alternatives relatively to a number of criteria which are conflicting in nature.

MULTIAGENT SIMULATOR

In our previous work we identified the main agents involved in a simulation of a group decision meeting (Marreiros et al. 2006; Marreiros et al. 2005): Participant Agents; Facilitator Agent; Register Agent; Voting Agent and Information Agent. Figure 1 illustrates the multi-agent model.

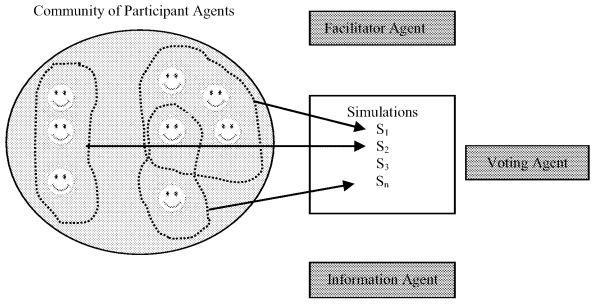


Figure 1 - Multi-agent model to simulate group decision

The Facilitator Agent caters for the simulation tasks in its organization (e.g. in decision problem and decision rules configuration). The Facilitator Agent also administrates the group formation process (i.e., the selection of the Participant Agents) and manages the inclusion of new Participant Agents into the agent community or agency. During the simulation, the facilitator agent coordinates and summarizes the simulation results.

Based on experience, it is also known that almost all the group decision making meetings have one or more voting rounds. The Voting Agent executes tasks related with the voting simulation process, according to the decision rules settled by the Facilitator Agent.

The Information Agent holds and retrieves information about the different proposals (or alternatives) that are evaluated by the agency during the group decision simulation process.

The Participant Agents also act according to the role usually fulfilled by the humans in the group decision making process. The set of Participant Agents set a community where the agents are created with social and emotional attributes that personalize their behaviour.

In the remain text of this section we will first present the architecture of participants agents, because they represent the main role in group decision making and then we will briefly detail the components of this architecture (Figure 2).

In the knowledge layer the agent has information about the environment where he is situated, about the profile of the other participant's agents that compose the simulation group, and regarding its own preferences and goals (its own profile). The information in the knowledge layer is dotted of uncertainty and will be accurate along the time through interactions done by the agent.

The interaction layer is responsible for the communication with other agents and by the interface with the user of the group decision making simulator.

The reasoning layer contains three major modules:

- The argumentative system – that is responsible for the arguments generation. This component will generate explanatory arguments and persuasive arguments, which are more related with the internal agent emotional state and about what he thinks of the others agents (based on a profile, where is included the emotional state);
- The decision making module – will support agents in the choice of the preferred alternative and will classify all the set of alternatives in three classes: preferred, indifferent and inadmissible;
- The emotional system (present in Marreiros et al. 2006) – will generate emotions and moods, affecting the choice of the arguments to send to the others participants, the evaluation of the received arguments and the final decision.

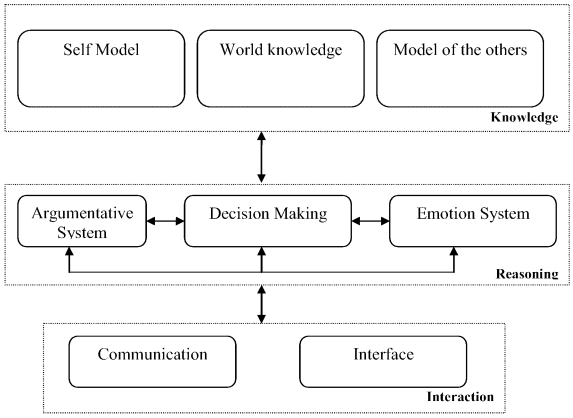


Figure 2– Participant agent architecture

ARGUMENTATION SYSTEM

This component will generate persuasive arguments based on the information that exists in the Participant Agent knowledge base. We adopt the same ontology as in (Kraus et al. 1996). So, we have the following arguments: appeal to prevailing practice; a counter example; an appeal to past promise; an appeal to self-interest; a promise of future reward; and a threat.

In this section we start by presenting the participant agent knowledge base and its strategies, followed by the presentation of the argumentation protocol for two agents and the description of the persuasion process (i.e. selection of the opponent, arguments generation, selection and evaluation).

Participant Agent Knowledge Base

The Participant Agents Knowledge Bases (KB) are made of logic clauses of the form $r_k: P_{i+j+1} \leftarrow P_1 \wedge P_2 \wedge \ldots \wedge P_{i-1} \wedge \text{not } P_i \wedge \ldots \wedge P_{i+j}$, where i, j, $k \in N_0$, $P_1, \ldots, P_{i+j}$ are literals; i.e., formulas of the form p or $\neg p$, where p is an atom, $\neg$ stands for explicit negation and where r_k, not, P_{i+j+1}, and $P_1 \wedge P_2 \wedge \ldots \wedge P_{i-1} \wedge \text{not } P_i \wedge \ldots \wedge P_{i+j}$ stand, respectively, for the clause's identifier, the **negation-by-failure** operator, the rule's consequent, and the rule's antecedent. If i=j=0 the clause is called a **fact** and is represented as $r_k: P_1$.

This work is supported by the developments in (Neves 1984) where the representation of incomplete information and the reasoning based on partial assumptions is studied, using the representation of null values to characterize abnormal or exceptional situations.

A meta theorem-solver for incomplete information, represented by the signature demo:T,V→{true,false}, infers the valuation V of a theorem T in terms of false, true and unknown according to the following set of productions:

demo(T,true) ← *T.*
demo(T,false) ← ¬*T.*
demo(T,unknown) ← *not T,*
not ¬ *T.*

The Knowledge Base of a Participant Agent KB_{AgP} is:
KB_{AgP}= AgP_iO U AgP_iOO U AgP_iP U AgP_iPO, where AgP_iO are the goals of agent AgP_i, AgP_iOO are the set of goals that AgP_i believes the other agents hold, AgP_iP contains the model of its own profile and AgP_iPO contains what AgP_i believes about the other agents profile.
The AgP_iPO is defined according to a set of characteristics enumerated afterward:

- Emotional state – characterizes the mood of the agent, and can be positive, negative or neutral;
- Benevolence – an agent can be, or not, benevolent. If an agent is benevolent, and receive a request from another agent to change his preferred alternative to alternative X, he will accept the request without the need to support arguments, since the alternative X belong to his internal set of preferred alternatives. If a benevolent agent receives more than one request to change his preference, and all the requests are acceptable, he will accept the request of the most credible agent;
- More/less preferred argument – agents may have a specific preference about the arguments to send.
- Gratitude debts of the agent – result from previous interactions (simulations) in the community of participant agents (Novais et al. 2005);
- Enemies – agents that, for some particular reason, our participant agent does not like to interact;
- Credibility - each agent evaluates the credibility of other agents based on two key components: trustworthiness and expertise (Andrade et al. 2005).

Participant Agent Strategies

During its lifetime and particularly in a specific simulation agents behave in order to accomplish their goals. The goals of each participant agent (AgP_iO) are defined according to the agent strategies. In the model that we are proposing participant agents have two distinct strategies: behaviour dependent and goal dependent.
In the goal related strategy two tactics are defined influencing the agent argumentation dialogues, namely:

- Defending the most preferred alternative;
- Avoiding that an unacceptable alternative becomes the group solution.

The goal related strategy is influenced by another strategy that is the time dependent behaviour. If an agent is time-dependent and supposing that he is arguing to defend his preferred, but he concludes that it is impossible to achieve an agreement on that alternative, he will change his tactic to avoid unacceptable alternatives.
The behaviour related strategy determines the method for the selection of the participant to persuade (opponent):

- Determined – the agent will consume all the available arguments with a specific agent beforehand to other agent;
- Anxious – an agent using this strategy will try to persuade a specific agent, but if that agent refuses the persuasion he will move forwards to try to persuade another agent.

Strategies will influence agent's behaviour, namely the argumentation flow and certainly the simulation results.

Argumentation Protocol

During a group decision making simulation, participants' agents may exchange the following locutions: request, refuse, accept and request with argument.
Request (AgP_i, AgP_j, α, *arg*) - in this case agent AgP_i is asking agent AgP_j to perform action α, the parameter *arg* may be void and in that case it is a request without argument or may have one of the arguments specified in the end of this section.
Accept (AgP_j, AgP_i, α) - in this case agent AgP_j is telling agent AgP_i that it accepts its request to perform α.
Refuse (AgP_j, AgP_i, α) - in this case agent AgP_j is telling agent AgP_i that it cannot accept its request to perform α.
In Figure 3, it is possible to see the argumentation protocol for two agents. However, note that this is the simplest scenario, because in reality, group decision making involves more than two agents and, at the same time AgPi is trying to persuade AgPj that, this agent may be involved in other persuasion dialogues with other group members.

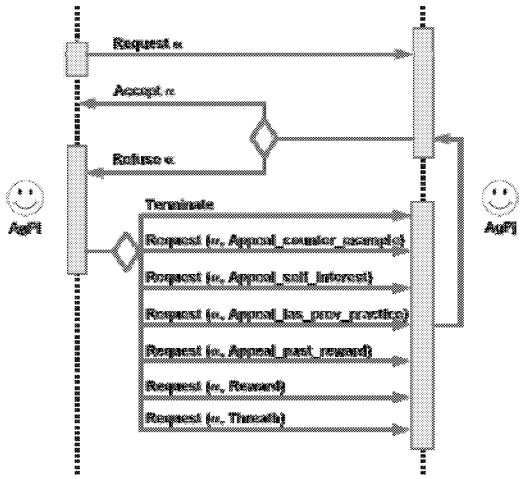


Figure 3 - Argumentation protocol for two agents

As we previously refer we adopt the same ontology as in (Kraus et al. 1996).. So, we have the following arguments: appeal to prevailing practice; a counter example; an appeal to past promise; an appeal to self-interest; a promise of future reward; and a threat. These are the arguments that agents will use to persuade each other.

Opponent Selection

A first step in an argumentation dialogue is the opponent selection. In this process the participant agent considers three different aspects, namely its own strategies (behaviour and goal related), the other agents voting preferences and the quality of the information detained about the others group members.

For evaluating the quality of the information, the following quality operators are defined: $Q_{EmotionalState}$, $Q_{Gratitude}$, $Q_{Credibility}$, $Q_{Enemies}$, $Q_{Benevolent}$, $Q_{TimeDependent}$ and $Q_{PreferredArguments}$.
where the quality of the information about the property K is given by: $Q_K = 1/Card$, Card is the cardinality of the exception set for K.
The quality of the information that agent AgPi detains about agent AgPj is measured by the following formula:

$$Q^{AgP_i}(Profile_{AgP_j}) = \frac{\sum_{k=1}^{N} Q_k^{AgP_j} * W_k^{AgP_i}}{\sum_{k=1}^{N} W_k^{AgP_i}}$$

where N is the number of properties of the profile, Q_k^{AgPj} is the quality measure of K and $W_k^{AgP_i}$ (in interval [0,1]) represents the contribution of K (weight) in the agent profile construction. A property with weight 0 will not be considered in the profile construction process.

Arguments Generation

In this section we describe and specify the persuasive arguments that our agents may change. For each argument it is given a little example.

Threats
As previously referred, threats are very common in human negotiation, and they can assume two distinct forms: you should perform *action A* otherwise I will perform *action B*; and you should **not** perform *action A* otherwise I will perform *action B.*
In our model this type of arguments may be formalized as a triplet:
Threat(Justification, Conclusion, Threatened_goal)

Example 1: AgP_1 ask AgP_2 to vote on alternative A_i with the argument that if he refuse he will vote on alternative A_j that he believes is unacceptable for AgP_2.

Promises
Like threats we can also have two distinct forms of rewards: If you perform action *A* I can perform *B*; and If you do not perform action *A* I perform *B*.
In our model this type of arguments may be formalized as a triplet:
Promise(Justification, Conclusion, Promised_goal)

Example 2: AgP_1 ask AgP_2 to vote on alternative A_i with the argument that if he accept he will stay in debt of gratitude.

Appeal to counter-example
In this case, the participant agent that makes a request supported by this argument, expect to convey the opponent that there is a contradiction between what he says and his past actions.
The argument appeal to a counter-example is an explanatory argument and in our model is formalized as:
Appeal_counter_example (Justification, Conclusion)

Example 3: AgP1 asks AgP2 to vote on alternative Ai, if AgP2 refuses then AgP1 may counter argument with an appeal to a counter-example saying for instance that in the past he preferred alternative Ak and change to Al, so why not to do the same now?

Appeal to self-interest
In this case the participant agent that makes a request supported by this argument expects to convince his interlocutor that making action *A* is of his best interest.
The argument appeal to self-interest is an explanatory argument and in our model is formalized as:
Appeal_self_interest (Justification, Conclusion)

Example 4: Suppose that AgP1 ask AgP2 to avoid voting on alternative Ai, supported by the argument that AgP3 voted in the Ai alternative and AgP2 do not like AgP3.

Appeal to past reward
In this case the participant agent that sends such kind of argument expects that his interlocutor performs an action based on a past promise.
In our model the appeals to past rewards are formalized as:
Appeal_past_reward (Justification, Conclusion)

Example 5: If in some point in the past agent AgP1 sent a request to AgP2 to vote on a specific alternative with the promise that he will stay with a debt of gratitude, if AgP2 accept it, in a future decision AgP2 can send a request supported by this argument.

Appeal to prevail practice
In this case, a participant agent believes that the opponent agent will refuse to perform a request action since it contradicts one of its own goals. For that reason the participant agent sends a request with a counter-example from third agent's actions or from past actions of the opponent.
In our model the appeal to prevailing practice is formalized as:
Appeal_past_reward (Justification, Conclusion)

Example 6: Suppose that AgP_1 knows that AgP_2 had a strong preference for the alternative A_i and changed for alternative A_j. If he intends to ask AgP_3 to change his preference to alternative A_j he could support his request with the argument that another agent, who also prefers alternative A_i, to change his preference to alternative A_j.

The last four types of arguments are explanatory arguments and are formalized as tuple:
Argument_type(Justification, Conclusion).

Arguments Selection

In (Kraus et al. 1996). it is used an existent pre-order for the selection of arguments to send, the strongest argument is a threat and the weakest argument is an appeal to prevailing practice. Another approach that can be finding in literature to govern the argument selection process is based on mixture of the alternatives utility and the trust in the interlocutor.

In our model it is proposed that the selection of arguments should be based on agent emotional state. We propose the following heuristic:
If the agent is in a good mood he will start with a weak argument;
If the agent is in bad mood he will start with a strong argument.
We adopt the scale proposed in (Walton and Krable 1995) for the definition of strong and weak arguments. We defined two distinct classes of arguments, namely a class for the weaker ones (i.e., appeals) and a class for the remainders (i.e., promises and threats). Inside each class the choice is conditionally defined by the existence in the opponent profile of a (un)preference by a specific argument. In case the agent does not detain information about that characteristic of the opponent, the selection inside each class follow the order defined in (Walton and Krable 1995).

Arguments Evaluation

In each argumentation round the participant agents may receive requests from several partners, and probably the majority is incompatible (e.g., in the same round the agent AgPi may receive a request to vote on alternative Ai and other to vote on alternative Aj). The agent should analyse all the requests based on several factors, namely the proposal utility, the credibility of proponent and the strength of the argument.
If the request does not contain an argument, the acceptance is conditioned by the utility of the request for the self, the credibility of the proponent and one of its profile characteristics, i.e., benevolence. Next is presented the algorithm for the evaluation of this type of requests (without arguments):

$$Req^{t}_{AgP_i} = \{request^{t}_{1}(AgP, AgP_i, Action), ..., request^{t}_{n}(AgP, AgP_i, Action)\}$$

, where AgP represents the identity of the agent that perform the request, n the total number of request received in instant t and Action the request action (e.g., voting on alternative number 1).

Begin
If $\neg$ *profile*$_{AgPi}$*(benovolent)* ***then***
Foreach
$request(Proponent, AgP_i, Action) \in Req^{t}_{AgP_i}$
refuse (Proponent, AgP$_i$, Action)
Else
Foreach
$request(Proponent, AgP_i, Action) \in Req^{t}_{AgP_i}$
If $AgPO_{AgP_i}$ $\square$ $Action$ ***then***
$Requests \leftarrow Requests \cup request(Proponent, AgP_i, Action)$
Else
refuse (Proponent, AgP$_i$, Action)

$(AgP, Requested_Action) \leftarrow Select_more_credible(Requests)$
Foreach $request(Proponent, AgP_i, Action) \in Requests$
If *(Proponent=AgP or Request_Action=Action)* ***then***
accept (Proponent, AgP$_i$, Action)
Else
refuse (Proponent, AgP$_i$, Action)
End

IMPLEMENTATION

Some details of the implementation of the simulator are described here. The system was developed in Open Agent Architecture (OAA), Java and Prolog. OAA is structured in order to: minimize the effort involved in the creation of new agents, that can be written in different languages and operating on diverse platforms; encourage the reuse of existing agents; and allow for dynamism and flexibility in the formation of agent communities. More information about OAA can be found in www.ai.sri.com/~oaa/.
Figure 5 shows the collection of agents that work at a particular moment in the simulator: ten participant agents, the facilitator agent (responsible for the follow-up of all simulations), the voting agent, the clock agent (OAA is not specially designed for simulation, for that reason it was necessary to introduce a clock agent to control the simulation), the oaa_monitor (i.e. an agent that belongs to the OAA platform, and is used to trace, debug and profile communication events for an OAA agent community) and the application agent (responsible for the communication between the community of agents and the simulator interface).

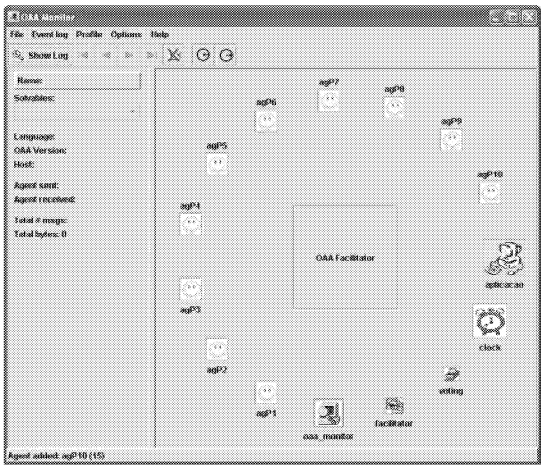


Figure 4- Community of agents

TEST

As referred earlier, the application area of simulator is the multi-criteria problem solving domain, dealing with problems more or less complex involving decisions that can be more or less polemic.

Table 1 - List of vacations options

Alternative	Cost	Relax	Adventure	Amusement
1	500	100	0	20
2	600	20	50	30
3	700	70	100	20
4	800	60	20	80
5	900	30	60	70

In this section it is described a small and simple scenario to illustrate the use of the multi-agent. The scenario used is trivial to enable a better understanding of the simulator: four

colleagues have to agree on a vacation destination. In table I it is possible to evaluate the different alternatives, characterized by a set of criteria (cost, relax, adventure, amusement).

In figure 5 it is shown the general configuration of the simulation and in figure 6 it is possible to see the emotional configuration process, in this particular case the user of the simulator only configured four types of emotion (hope, gratitude, fear and disappointment), two positive and two negative ones.

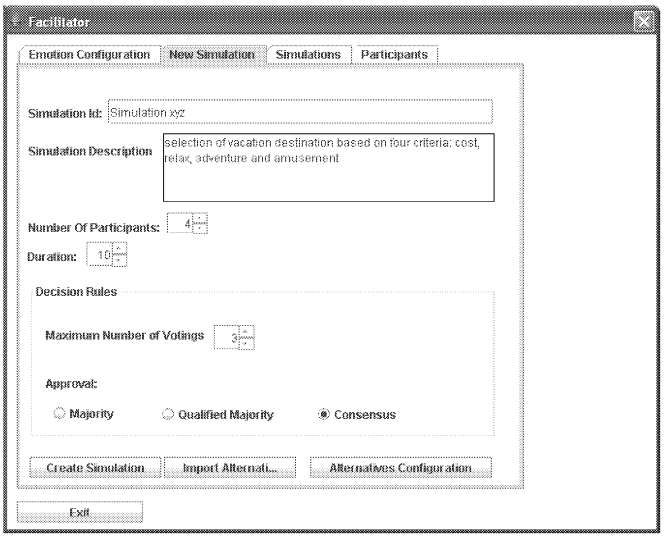


Figure 5 – Simulation configuration

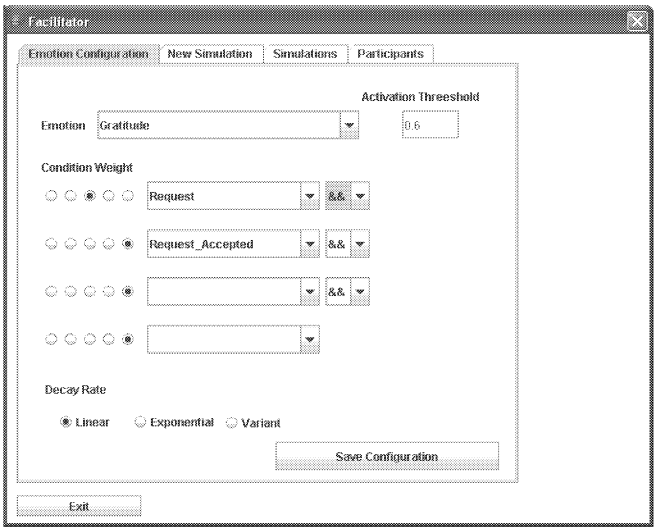


Figure 6 - Gratitude configuration

Figure 7 shows the profile of a particular agent.

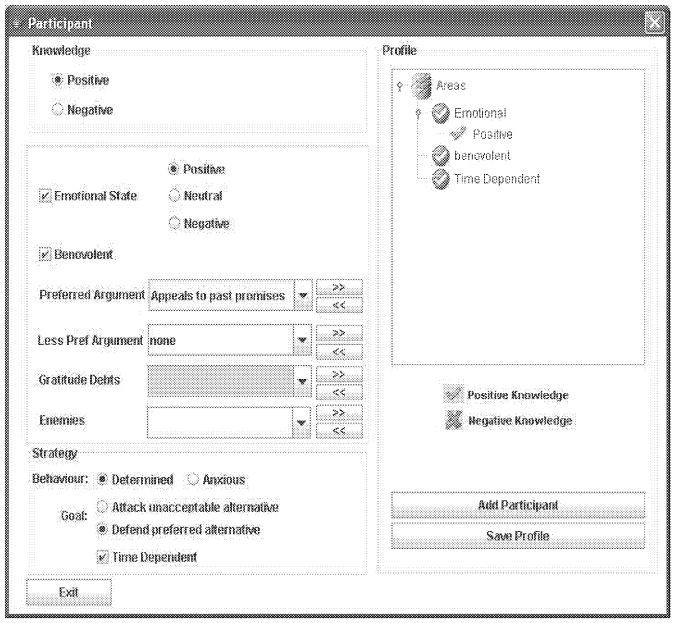


Figure 7- Agent Profile

The simulator user establishes the initial preferences of each agent. After, each agent divides and classifies the set of alternatives in three subsets: the most preferred ones, indifferent ones and unacceptable ones.

For a single illustration, one's can take a look to Figure 8. Agent AgP_1 and agent AgP_2 try to persuade AgP_4 to choose the 3rd alternative, appealing to self interest; AgP_4 tries to convince AgP_1 based on a previous simulation where initially AgP_1 prefers a specific alternative and then changes its preferences; AgP_3 tries to persuade its opponent (AgP_1) appealing to a past promise. In this case, a gratitude debt is involved. AgP_4 refuses all the received requests and AgP_1 refuses the request of AgP_3 and accepts the request AgP_4.

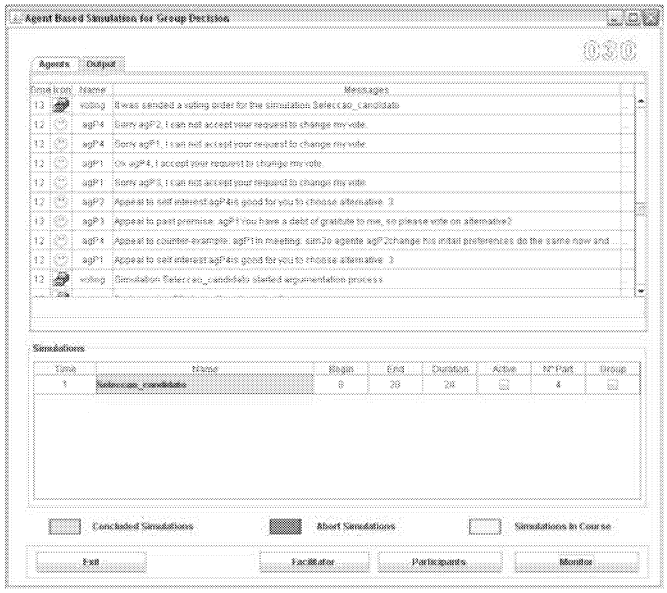


Figure 8 – Argumentation round

Based on the problem configured in table 1 and on variations of the agent's initial preferences, several scenarios were established in order to understand if emotional agents have more success in the simulations than non-emotional ones. The success rate of the different strategies is also evaluated.

Agents used in the experiences shows a behaviour that varies according to the strategy they subscribed, namely:

- Defending the most preferred alternative;
- Avoiding an unacceptable alternative;
- Mixing, starting the defence the most preferred strategy, but they may change, avoiding unacceptable ones if the agent is influenced by time.

One's consider that an agent succeeds:

- When implementing the "Defending preferred" strategy, the alternative strategy elected by the group is one of its "most preferred" set;
- When implementing "Avoiding unacceptable" strategy and the alternative selected by the group does not belong to it;
- When implementing the Mixing strategy and the elected alternative is one that is preferred or is indifferent for it.

In the group formation process,it was taken as pre-requisite that in each simulation all the agents subscribe the same strategy.

Figure 9 shows the success rate of participant agents (i.e. simulation success rate) and, particularly the subscribed strategies. It were taken in account all the performed simulations (all the variations constructed are based on table 1 and on the agent initial preferences).

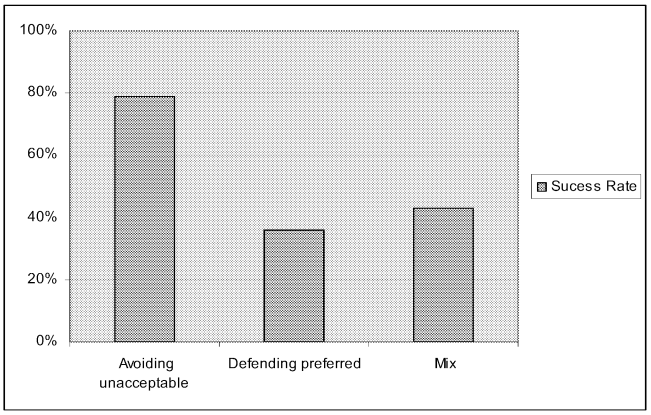


Figure 9 – Agents strategies and success rate

The experiments made with emotional and non-emotional agents showed that the number of exchanged arguments to achieve an agreement is less when emotional agents are used,

CONCLUSIONS

It is proposed a multi-agent simulator of group decision making. Agents exhibit different emotional states and deal with incomplete information, either at the representation level, or at the reasoning one. Each agent represents a heterogeneous group member, in the behavioural sense. The discussion process between group members (or agents) is interpreted through the exchange of persuasive arguments, built around the same premises stated before. A framework has been developed to support participants during meetings, enabling the previewing of the argumentation results, testing and gaging strategies. It is a kind of "what-if" performance, dealing with the argumentation phase of group decision making. Future work will enhance the system with the inclusion of profiles, tracing the participant's performance in order to verify its context awareness.

ACKNOWLEDGEMENTS

This work was partially supported by ArgEmotionAgents project (POSI/EIA/56259/2004 - Argumentative Agents with Emotional Behaviour Modelling for Participants' Support in Group Decision-Making Meetings), which is a project supported by FCT (Science & Technology Foundation – Portugal).

REFERENCES

Amgoud, L., Belabbès, S., Prade, H. 2006, A Formal General Setting for Dialogue Protocols, Lecture Notes in Computer Science, 4183, pp. 13-23.

Andrade F.; J. Neves; P. Novais; J. Machado; and A. Abelha. 2005, "Legal Security and Credibility in Agent Based Virtual Enterprises", in Collaborative Networks and Their Breeding Environments, Camarinha-Matos L. Afsarmanesh H., Ortiz A., (Eds), Springer-Verlag, ISBN 0-387-28259-9, pp 501-512.

GroupSystems –URL: www.groupsystems.com

Ito, T. and T. Shintani. 1997, "Persuasion among agents: An approach to implementing a group Decision Support System based on multi-agent negotiation", Proceedings of the 5th International joint Conference on Artificial Intelligence.

Jennings, N. Parsons, S. Noriega, P. and Sierra, C. 1998, On Argumenta-tion-Based Negotiation. Proceedings of the International Workshop on Multi-Agent Systems, USA.

Karacapilidis, N. and D. Papadias. 2001, "Computer supported argumentation and collaborative decision making: The Hermes system", Information Systems, Vol. 26 No. 4 pp. 259-277.

Kraus, S., Sycara, K. and Evenchick, A. 1998, Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 104(1-2), pp 1-69.

Kudenko, D.; M. Bauer; and D. Dengler. 2003. "Group decision making through mediated discussions". Proceedings of the 10th International conference on user modelling.

Marreiros G., Novais P., Machado J., Ramos C., Neves J. 2006, Emotions on Agent Based Simulators For Group Formation. Proceedings of the ESM 2006 - The 20th annual European Simulation and Modelling Conference, Toulouse, France, ISBN 90-77381-30-9, pp 310-314.

Marreiros, G. Ramos, C. and Neves, J. 2005, Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision, Lecture Notes in Computer Science, 3823 pp. 41-50.

Neves, J. 1984, A Logic Interpreter to Handle Time and Negation in Logic Data Bases. Proceedings of ACM'84, The Fifth Generation Challenge, pp. 50-54.

Novais P., Brito L., Neves J. 2005, Pre-Argumentative Reasoning, The Knowledge-Based Systems Journal, vol. 18 Nº 2-3, pp 79-88, ISSN 0950-7051, Elsevier Science BV.

Payne, T.; T.L. Lenox, S. Hahn; M. Lewis; and K. Sycara. 2000. "Agent-Based Team Aiding in a Time Critical Task", Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE Press, Los Alamitos.

Sierra, C.; Jennings, N.R.; Noriega,P.; Parsons, S. 1998. A framework for argumentation-based negotiation. Intelligent agents IV LNAI, Vol.1365 pp. 177-192.

Turban, E. and Aronson, J. 1998, Decision support systems and intelli-gent systems. Prentice-Hall, Inc.

Walton, D. and Krabbe, E. 1995, Commitment in Dialogue. Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY.

Zamfirescu, C.B.; C. Candea; and S. Luca. 2001. “On Integrating Agents into GDSS”, Proceedings of the 9th IFAC/IFORS/IMACS/IFIP Conference on Large Scale Systems, Bucharest, pp. 231-236.

AUTHOR BIOGRAPHIES

GORETI MARREIROS is an assistant professor of Computer Engineering at the Institute of Engineering – Polytechnic of Porto (ISEP/IPP) and researcher at the Knowledge Engineering and Decision Support Research Group (GECAD). She is preparing her PhD thesis on multi-agent simulation to support argumentation and group decision making. She received her MSc in Information Management from the University of Porto in 2002.

PAULO NOVAIS is a Professor of Computer Science at the University of Minho, Braga, Portugal. He received a PhD in Computer Sciences from the same university in 2003. He is Vice-president of APPIA, the Portuguese Association for Artificial Intelligence. His current research directions span the fields of Knowledge Representation and Reasoning Systems, Agent and Multi-agent Systems, Ambient Intelligence and Collaborative Networks. His web page is available at http://www.di.uminho.pt/˜pjn.

JOSÉ MACHADO is a Professor of Computer Science at at the University of Minho, Braga, Portugal. He received a PhD in Computer Sciences from the same university in 2002. His current research directions span the fields of Agent and Multi-agent Systems, Knowledge Representation and Reasoning Systems and Ambient Intelligence.

CARLOS RAMOS is a Coordinator Professor of Computer Engineering and director of the Knowledge Engineering and Decision Support Research Group (GECAD) at the Institute of Engineering – Polytechnic of Porto (ISEP/IPP). His research interests are Artificial Intelligence and Decision Support Systems. He received his PhD in Electrical and Computer Engineering from the University of Porto in 1993.

JOSÉ NEVES is Full Professor of Computer Science at the University of Minho. He got his PhD in Computer Science from Harriet Watt University, Edinburgh, Scotland in 1984. His current research activities span the fields of Extended Logic Programming, Knowledge Representation and Reasoning systems and Multi-agent Systems.

TECHNICAL DIFFUSION BY SOCIAL MEANS: AN AGENT BASED FRAMEWORK FOR DISTRICT HEATING

Marco Remondino
Department of Computer Science
University of Turin
Italy
E-mail: remond@di.unito.it

KEYWORDS
Social Networks, Software Agents, Simulation, Diffusion.

ABSTRACT

The spreading of certain innovations can't be modelled just by taking into accounts pure technical issues; sometimes the forecasts made with a statistical or process based model, which considers just the engineering or financial aspects (i.e.: production process, activity decomposition, scale economies, product life cycle and so on), prove to be inaccurate or even wrong. This is commonly the case of cellular telephones, air-cooling systems, personal computers and so on, fields in which some external influences have led to diffusion results which are very different from the initial previsions obtained with classical prevision tools. The purpose of this work is to examine the diffusion of an emerging technical innovation: the district heating. The spreading of this technology shouldn't be affected by phenomena like fashion – as happened with cellular phones - and its adoption can't be considered a status symbol. Although, there are some other influences which can't be taken into account by a traditional process based or statistical simulator, i.e.: the effects of individual perception about savings, ecological impacts and so on.

INTRODUCTION

District heating is a system for distributing heat generated in a centralized location for residential heating requirements. The heat is often obtained from a cogeneration plant; more generally district heating is the name assigned to the creation of an infrastructure with pipes, buried in the ground and filled with warm water, to satisfy concentrated heat requirements on local heat markets. Heat is normally supplied to the pipes by retrieving and refining heat that other entities have no further use for and by using fuels or natural heat that end-users cannot use easily.

Examples of such heat sources are heat retrieved from thermal power processes (also known as combined heat and power), industrial processes and refuse incineration plants as well as geothermal heat and wood waste. District heating is normally delivered to satisfy heat requirements in cities for heating and hot water in residential, commercial, and public buildings. The heat requirements must be concentrated in order to attain low distribution costs.

This technology is diffused in north European countries and is also spreading to the rest of Europe; in particular the analysis in this paper is conducted with the support of AEM – Iride Turin, i.e.: the electrical company of Turin, a city located in north-western Italy.

A technical and statistical analysis can give hints about how the district heating is spreading and will spread on the territory, but can't answer to some very important questions, i.e.: how can incentives and benefits increase the adoption of district heating? How can phenomena like word-spreading, or the perceived convenience, or the concerns about environment and nature influence the diffusion of this technology?

In this paper an agent based framework is described, which should address such questions and others, and that could be applied to other similar technologies, like solar photovoltaic panels and so on.

AGENT BASED MODELING AND SIMULATION

Agent Based Modelling is the most interesting and advanced approach for simulating a complex system: in a social context, the single parts and the whole are often very hard to describe in detail. Besides, there are agent based formalisms which allow to study the emergency of social behaviour with the creation and study of models, known as artificial societies. Thanks to the ever increasing computational power, it's been possible to use such models to create software, based on intelligent agents, which aggregate behaviour is complex and difficult to predict, and can be used in open and distributed systems.

In Franklin and Graesser (1997) we read that:

An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

Another very general, yet comprehensive one is provided by Jennings (1996):

...the term [agent] is usually applied to describe self-contained programs which can control their own actions based on their perceptions of their operating environment.

Agents themselves have traditionally been categorized into one of the following types (Woolridge and Jennings, 1995):

- Reactive
- Collaborative/Deliberative
- Hybrid

When designing any agent-based system, it is important to determine how sophisticated the agents' reasoning will be. Reactive agents simply retrieve pre-set behaviours similar to reflexes without maintaining any internal state. On the other hand, deliberative agents behave more like they are thinking, by searching through a space of behaviours, maintaining internal state, and predicting the effects of actions. Although the line between reactive and deliberative agents can be somewhat blurry, an agent with no internal state is certainly reactive, and one that bases its actions on the predicted actions of other agents is deliberative.

The agents used in this paper are reactive, but organized in the form of a MAS (multi agent system), which can be thought of as a group of interacting agents working together or communicating among each other. To maximize the efficiency of the system, each agent must be able to reason about other agents' actions in addition to its own. A dynamic and unpredictable environment creates a need for an agent to employ flexible strategies. The more flexible the strategies however, the more difficult it becomes to predict what the other agents are going to do. For this reason, coordination mechanisms have been developed to help the agents interact when performing complex actions requiring teamwork. These mechanisms must ensure that the plans of individual agents do not conflict, while guiding the agents in pursuit of the goals of the system..

SOCIAL NETWORKS AND CONNECTIONS

A social network is defined as "a set of nodes - e.g. persons, organizations - linked by a set of social relationship - e.g. friendship, transfer of funds, overlapping membership - of a specific type" (Laumann, et al., 1978).
Any kind of network can be described in terms of a graph, composed of nodes and a set of lines, edges, joining the nodes. In a mathematician's terminology, a graph is a collection of points and lines connecting some (possibly empty) subset of them. The points of a graph are most commonly known as graph vertices, but may also be called nodes or simply points. Similarly, the lines connecting the vertices of a graph are most commonly known as graph edges, but may also be called arcs or lines.

The study of graphs is known as graph theory, and was first systematically investigated by D. König in the 1930s (Gardner, 1984). Graphs come in a wide variety of different sorts. The most common type is graphs in which at most one edge (i.e.: either one edge or no edges) may connect any two vertices. Such graphs are called simple graphs and are the ones we'll use in our analysis. The edges of graphs may also be imbued with directedness. A normal graph in which edges are undirected is said to be undirected. Otherwise, if arrows may be placed on one or both endpoints of the edges of a graph to indicate directedness, the graph is said to be directed. The connections here represent the level of "friendship", and serve the purpose of transmitting the opinions about the new technology.

In the present work the graph used to represent the social network linking the agents together is bi-directed, i.e. each edge points on both directions as once. This seems realistic, since we can imagine our network as a group of people who know each other. If A knows B, then it's quite obvious that B knows A in turn; we don't voluntarily consider those situations in which a subject disseminates her/his opinion to others and isn't touched by their decisions (e.g.: advertisement, political campaigns, and so forth). That's because we suppose that this sort of dissemination comes "a priori", i.e. before our analysis starts; we are now interested in studying how a set of agents mutually connected into a network can influence one another and come to a final overall result.

The only exception is that of some "special" agents, i.e.: the property managers, who are connected to the other agents via a mono-directional link (since they are supposed to spread their knowledge towards the people living in the condominiums, but not to be affected in turn by their opinions).

THE SIMULATION FRAMEWORK

A number n of macro-agents is built, that represent the condominiums; each of those is composed of a random number of agents (where the range can be set with min/max or can obey to a normal distribution, given an average).

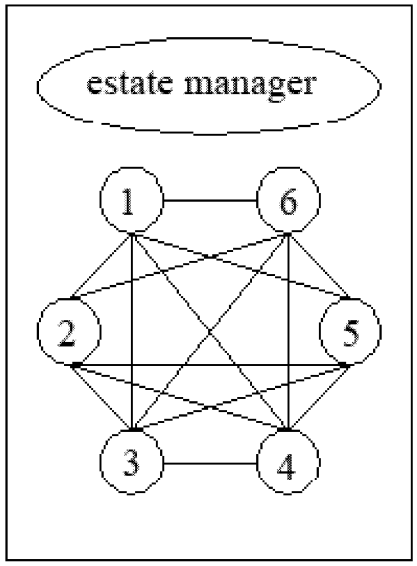


Figure 1: the Condominium and the Agents living in it

The agents belonging to the same condominium will be connected by a network; the links will feature a random weight, with a modifiable threshold. The network has the purpose of transmitting the information about the technology,

that's coded through a function describing the perception of the real phenomena. The perception is surely derived from the real saving allowed by the new technology, but not only that; there could be a bias, representing other factors, like trustfulness towards the company, ecological reasons and so on. The agents will spread their own perception on the network, to the other agents connected to them and this will be modified according to the weight of the specific individual link. Each macro-agent will also feature a property/estate manager, who could also be shared by two or more condominiums, thus constituting a bridge among different macro-agents and, indirectly, among the agents involved.

A second link among the macro-agents is represented by social networks among the agents living in different condominiums. We can metaphorically depict these links as friendship, job or family relationship and so forth.

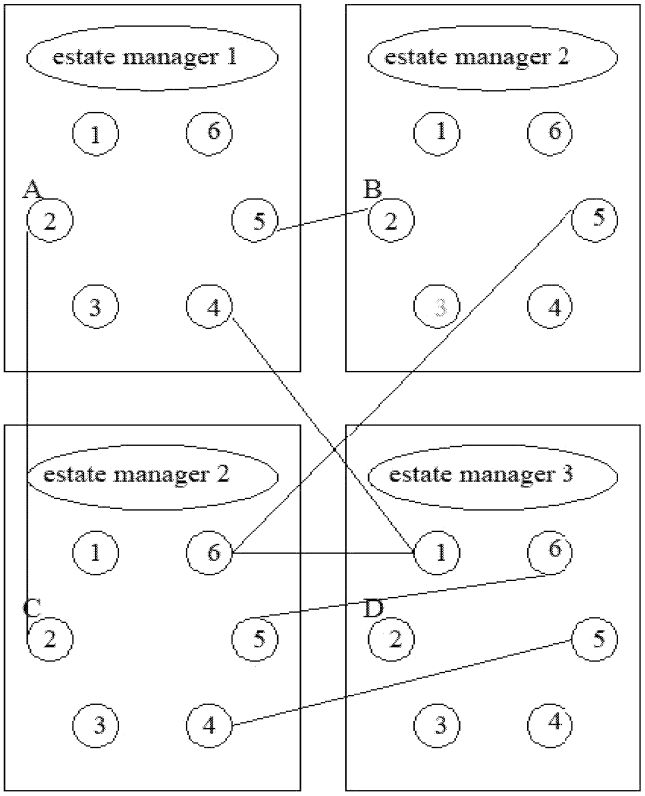


Figure 2: Different Condominiums Linked by Social Relations among Individuals

The important aspect is that only agents are connected, not the condominiums; agents represent people, so they can be connected somehow, while condominiums can be considered "connected" among them if and only if the people living in them know each other.

Since the focus is how the district heating can spread thanks to social relations among people, the model, through repeated execution, can show the diffusion over time, by changing just one parameter at a time, while leaving the others unchanged (*ceteris paribus*).

The Parameters in the Model

Here follows a list of the parameters used in the simulation:

1) Total number of condominiums
2) Average number of agents per condominium
3) Average or maximum weight for the links among the agents living in the same condominium
4) Percentage of condominiums already reached by the new technology
5) Cost for the installation (una tantum)
6) Average saving allowed by the new technology
7) Perception BIAS (1:double perception, 0:perfect perception, -1:inverse perception, 2:random perception)
8) Average or maximum weight for the links among the agents living in different condominiums
9) Satisfaction Threshold (0: always satisfied, 1: satisfied only if perfect conditions occur)
10) Requested majority to adopt the district heating for the condominiums (percentage)

These parameters can be changed by the user at the beginning of the simulation, in order to mimic, as much as possible, the real world. Some parameters are very general (such as "Average number of agents per condominium", "Average or maximum weight for the links among the agents living in the same condominium", "Cost for the installation" and so on). They should be set accorging to the values observed on samples taken in the city where we want to set the study.

Other parameters, such as "Satisfaction Threshold" and "Perception BIAS" are more difficult to determine; they are derived from the Cognitive Sciences and define how a single person perceives the environment and the innovations. It is possible to assume that almost nobody has a perfect perception, since this would be similar to the "Homo oeconomicus" assumption in the Neoclassical Economics, i.e.: a stylized and ideal type of the instrumentally rational individual. More realistically, people have a perception BIAS, that's the general tendency of a person to view a situation in a particular light based upon the method of socialization that they were brought up with, their personal experience, their knowledge and so on.
A technician could have almost a perfect perception about how convenient a technical innovation can be, but an average person can't.
Besides, not all the individuals are satisfied about something at the same threshold; somebody could be happy to save 100€ per year, while others don't even bother for that amount of money.
That's why these two parameters must be set according to some general observations and common sense.

Model Iterations and Steps

In this subsection an overview about how the model works is given. Here follows the sequence of steps performed during the simulation:

1) The first step is the adoption of the new technology by a number of condominiums
2) The new technology has some installation costs, and brings a positive or negative saving compared to historical data
3) These values will be perceived by the agents living in the condominiums, but filtered by their personal BIAS (which takes into account different income, ecological awareness and so on)
4) Each agent will spread through the social network her/his own perception, and the agents linked with her/him will form their own opinion, according to the weighted average of the received opinions
5) According to the "satisfaction threshold", the agents will individually decide whether to adopt or not the new technology
6) Once all the agents in a condominium have formed their own opinion, according to the requested majority, the condominium will adopt the new technology or not yet. The next step will be a loop from point number 2).

There is not a metaphorical correspondence between the steps and time units. Since in six months many condominiums can pass to the new technology, we could realistically think at heach step as one month time.

RESULTS AND COMMENTS

Several experiments have been run on the model, by changing the parameters in order to mimic different scenarios. Two are the main parameters that affect the diffusion over time: a different connection level and a different perception level. In particular, since we are dealing with a single city, we we can expect a saturation level, that's a threshold beyond which it's not possible to go (i.e.: a limited number of agents can be reached by district heating, since a limited number of people live in a city). So we aspect an higly connected network to increase the diffusion speed over time (we can think to small villages as an example).
This threshold is represented by the parameter called "Total number of condominiums" that multiplied per "Average number of agents per condominium" gives a good approximation of the total number of agents in the simulation (not exact, due to sampling effects). On the graphs, the new adoptions per time step are shown.

In figure 3, the results with a low perception index and a loosely connected network are shown, while in figure 4 there are the results with the same low perception but with an (unrealistically) fully connected network. As noticieable in the comparison in figure 5, the fully connected scenario leads to a faster adoption of the new technology, followed by a steady trend, when almost all the condominiums are reached.

In the next three figures (6, 7 and 8) the Fully Vs Loosely networks are tested in a high perception environment. Since the new technology brings positive savings, the scenario with a loosely connected network seems less noisy and more regular. The scenario with a fully connected network, on the other end, is that in which in few steps all the condominiums adopt the new technology, after which the adoption rate falls to zero.

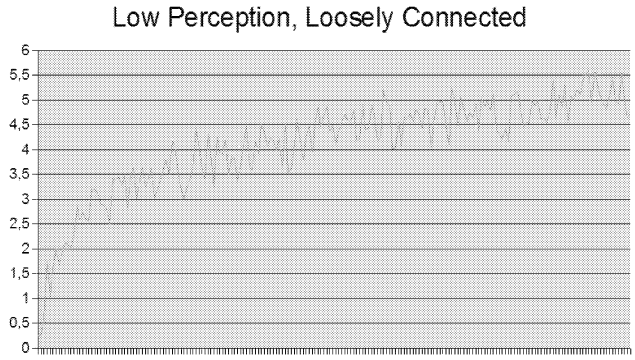


Figure 3: Low Perception and Loosely Connected Network

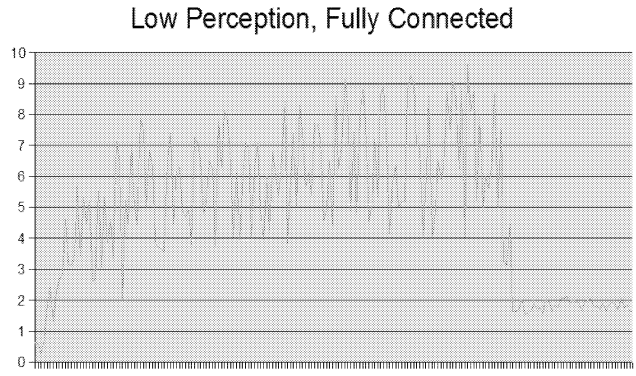


Figure 4: Low Perception and Fully Connected Network

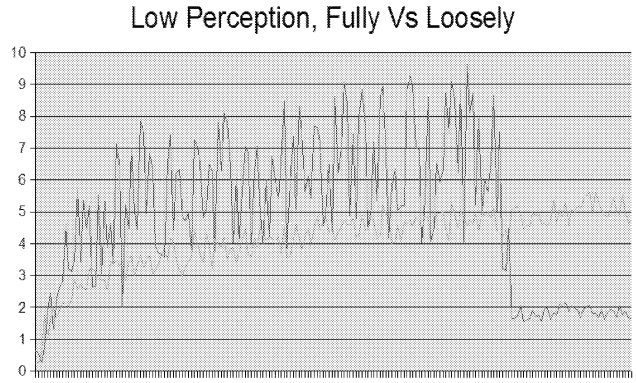


Figure 5: Comparison among Loosely and Fully Connected with Low Perception

Considering the real data from AEM Torino, the most realistic situation is the LP-LC one.
In the real-world, normalized data, there is a logarithmic-like growth of the diffusion, stabilizing at 5x the initial value and then slowly decreasing (when most of the city is reached).
In the real world, as to the district heating technology, the individual perception is thus to be considered lower than the real economical benefit; that's probably because there are not fashion issues connected with this technology.
Very different from this is the case concerning the spreading of the cellular phones (whose diffusion resembles the initial path of HP-FC curve in figure 7), also because, differently

from the case of district heating, a single person can change cellular phone over time, and thus the market won't saturate easily.

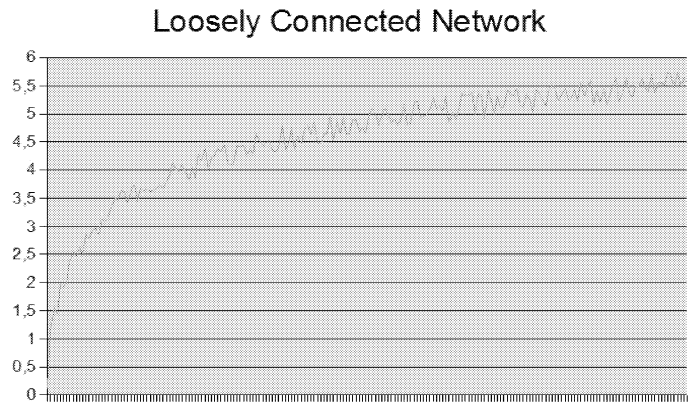


Figure 6: High Perception and Loosely Connected Network

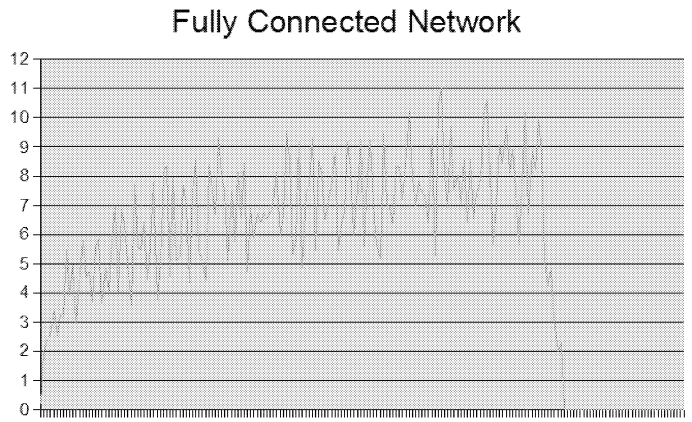


Figure 7: High Perception and Fully Connected Network

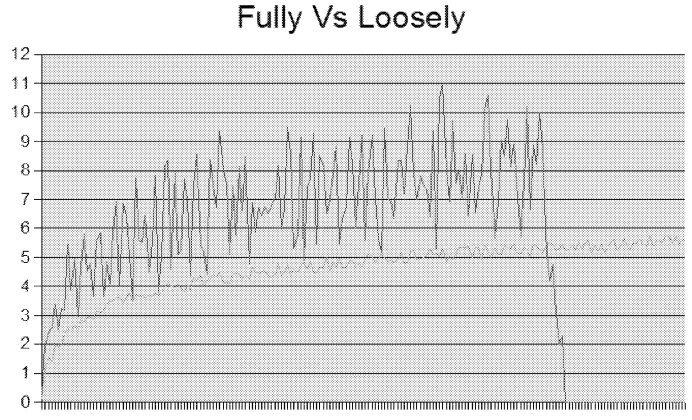


Figure 8: Comparison among Loosely and Fully Connected with High Perception

CONCLUSION

An agent based framework to test the social impact on technology diffusion has been presented; in it, a certain number of agents are connected by some sort of relation, by a social network. As a testbed for the model, the case of District Heating is analyzed: in it there is no fashion involved, but just economical decisions related to the money that can be saved with the new technology, over time.

The core parameters of the model are the level of perception, describing how well the real impact of the new technology is perceived by the agents, and the level of connection of the social network connecting the agents.

An empirical validation, comparing data coming from the simulation to real ones, shows that for district heating the perception is "low" and the social network is loosely connected. This means that some kind of innovations, like those in which fashion is not an important factor, spread slowly even if they are technologically relevant and bring savings.

REFERENCES

Franklin, S., & Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Proceedings of the Agent Theories, Architectures, and Languages Workshop, Berlin (pp. 193-206). Springer Verlag.

Jennings, N. R. (1996). Software agents. IEEE Review (pp. 17-20)

Laumann E.O. et al. 1978, "Community Structure of Interorganizational Linkages", Annual Review of Sociology, 4, pp 455-484

Mataric, M. (1995). Issues and approaches in the design of collective autonomous agents. Robotics and Autonomous Systems 16 (pp.321-331)

Woolridge, M., & Jennings, N. (1995). Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2) pp.115-152

AUTHOR BIOGRAPHY

MARCO REMONDINO got his Master Degree in Economics at the beginning of 2001, with 110/110 cum Laude et Mentione. He then started a PhD in Computer Science, during which he delved into theoretical studies about the structure of different software agents, namely the reactive and deliberative (BDI) ones. He completed his PhD in January 2005.

After that, he was awarded a two-year research scholarship from ISI Foundation, for the Lagrange Project on Complex Systems, during which he applied agent based paradigms to design several models in different scientific fields, namely Game Theory, Biology, Economics and Enterprise Simulation.

At present, he holds a post-DOC research grant from University of Turin, Department of Computer Science.

His main research interests are Agent Based Model-ling, different paradigms for agents and their integration, computer simulation, Social Systems, emergent properties for Complex Systems, Social Networks, Action Selection, agent learning through Neural Networks, Genetic Algo-rithms, Classifier Systems, paradigms of Reinforcement Learning, Data Mining, validation of models.

AGENT BASED MODELS

Agent-Based Model of Methicillin-Resistant *Staphylococcus aureus* and Antibiotics in Batch Culture.

James T. Murphy
Ray Walshe
Modelling & Scientific Computing Research Group,
School of Computing, Dublin City University,
Dublin 9, Ireland,
Email: {jamurphy, ray.walshe}@computing.dcu.ie

Marc Devocelle,
Centre for Synthesis and Chemical Biology,
Department of Pharmaceutical and Medicinal Chemistry,
Royal College of Surgeons in Ireland,
123 St. Stephen's Green, Dublin 2, Ireland.
Email: mdevocelle@rcsi.ie

KEYWORDS

Biology, Agent-based, Antibiotics, MRSA.

ABSTRACT

An agent-based software model called 'Micro-Gen Bacterial Simulator' has been developed to model the growth and development of Methicillin-resistant *S. aureus* (MRSA) bacteria. It can be used as a tool to investigate the connections between antibiotic-resistance mechanisms and their effects on population growth. By inputting low-level biochemical/kinetic parameters, it simulates the pharmacodynamics of MRSA cultures interacting with antibiotic molecules *in vitro*. Initial testing shows that it can predict the Minimum Inhibitory Concentration (MIC) of an antibiotic, which is the key measure of antibiotic efficacy used in the clinical setting. Micro-Gen represents a robust tool for facilitating the rapid screening of drug compounds against a wide array of bacterial strains in simulated culture conditions.

INTRODUCTION

The Micro-Gen Bacterial Simulator was developed to model the growth and interactions of bacterial cells with antibiotics *in vitro* (Murphy & Walshe, 2006). It is an agent-based model which simulates the interactions of individual bacterial cells and antibiotic molecules in a discrete two-dimensional environment. This allows for spatial heterogeneity in nutrient content and the distribution of bacteria, as opposed to assuming a completely homogeneous, mixed environment. Each bacterium is supplied with pre-defined rules and parameters, and the emergent properties of the population can be examined without the need for population-level (global) laws (Jennings et al., 1998).

Micro-Gen represents a robust platform on which to build a detailed model of Methicillin-Resistant *S. aureus* (MRSA) bacteria and their interactions with β-lactam antibiotics. It provides an important theoretical framework for analysing the antibiotic resistance strategies used by MRSA. By studying the effects of changes at the individual molecular and cellular level, it can give insights into the connections between resistance mechanisms and their effects on population growth.

The mediator of Penicillin resistance in *S. aureus* is an enzyme, called β-lactamase, which hydrolytically cleaves the β-lactam ring present in Penicillin, rendering the drug ineffective (Kotra & Mobashery, 1998). Methicillin was introduced in 1959 to treat infections of Penicillin-resistant *Staphylococcus aureus*. However, Methicillin-resistant *S. aureus* (MRSA) strains emerged soon after with a second mechanism of resistance (Fuda et al., 2005).

MRSA are characterized by alterations to the protein targets, called Penicillin-Binding Proteins (PBP), that the β-lactam antibiotics bind to in the bacterial cell membrane. PBPs mediate cell wall synthesis in bacteria, but when antibiotics bind to and acylate them they are unable to function correctly. MRSA bacteria produce a PBP (called PBP2a) with a decreased affinity for most β-lactam antibiotics, including Penicillin and Methicillin (Fisher et al., 2005). PBP2a is expressed in addition to the normal antibiotic-susceptible PBP2 protein and supplements its cell wall synthesis activity.

MODEL DESCRIPTION

The simulations described here were carried out using version 2 of the Micro-Gen Bacterial Simulator. Micro-Gen, which is coded in the C++ object-oriented programming language, implements an agent-based model of bacterial growth in laboratory culture conditions (Murphy & Walshe, 2006). It is an extended and modified version of the Java software tool BAIT (Bacteria-Antibiotic Interaction Tool) (Walshe, 2006). Version 2 of Micro-Gen includes significant improvements in performance and feature set over the original.

Environment

The culture environment is represented by a discrete, two-dimensional grid of 'patches' containing diffusible elements such as nutrients, enzymes and antibiotics which the bacterial cells interact with. The diffusion of molecules through the environment is governed by a discretized implementation of Fick's First Law of diffusion (Ginovart et al., 2002). According to this, the amount of substance exchanged by two adjacent patches is equal to the concentration difference multiplied by a user-defined diffusion coefficient. For diagonally-adjacent patches the diffusion rate is modified by the factor $1 / \sqrt{2}$.

The area of each patch represents 1 μm^2 of agar medium. Based on microscopic observations, the average diameter of a newly divided *S. aureus* cell is estimated to be 0.5 μm (Giesbrecht et al., 1998). Therefore, each patch can accommodate approximately four newly divided cells, before overcrowding comes into effect. When overcrowding occurs, a cell shunting algorithm is implemented that determines cell movement into adjacent patches. The direction a bacterial cell is shunted is determined by sampling from a probability distribution of the cell densities in surrounding patches.

MRSA Bacteria Agents

The individual bacteria of a colony are represented by software agents, which store the physical traits such as energy state or antibiotic damage and the behavioural rules of the bacteria (Murphy & Walshe, 2006). *S. aureus* have spherical-shaped, non-motile cells and a generation time of approximately 29 minutes (Ender et al., 2004). They follow the characteristic bacterial life cycle and reproduce asexually by the process of binary fission, whereby the cell divides into two identical daughter cells. In situations of nutrient deprivation, cells enter a stationary phase of reduced metabolic activity as a survival strategy.

At the start of the simulation, a 'fabric' (represented as an array) of bacterial objects is initialized. Bacterial agents in the fabric are flagged as either alive or dead, and when new agents are 'born', a free agent in the fabric is updated to 'alive' status and the relevant parameters updated. The use of a fabric data structure results in significant performance gains since new objects do not have to be created or destroyed in memory each time a bacterium reproduces or dies. The memory access time (which can be a significant bottleneck in large simulations) is also reduced by maintaining 'alive' agents in contiguous blocks of memory.

β-Lactamases
If the bacterial agents interact with antibiotics in the same patch, expression of β-lactamase enzyme is induced. This is released into the extra-cellular milieu where it is subject to the rules of diffusion described by Fick's law (see above). There is a finite period of time between initial interaction with antibiotic and full induction of β-lactamase expression (Lewis et al., 1999). This is incorporated into the model by including a time delay of eighty minutes between first antibiotic contact and full β-lactamase expression. During this time, there is an exponential increase in the β-lactamase production rate up to the full production level. When the antibiotic is removed from the immediate environment of the cell, re-repression of β-lactamase expression occurs. Free β-lactamase enzyme in the environment interacts with antibiotic molecules according to defined kinetic parameters (see below).

PBP2a
The key distinguishing characteristic of MRSA cells is the presence of PBP2a protein in the cell membrane, which is involved in cell wall synthesis. When β-lactam antibiotics interact with the bacterial cell their efficacy is limited by their ability to bind to and acylate the PBP2a protein. Parameters accounting for the binding affinity and acylation rate of PBP2a have been incorporated into the model.

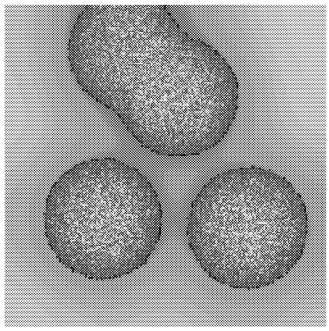


Figure 1: Screenshot from Micro-Gen of Simulated MRSA Colonies Growing on Nutrient Agar Medium

Kinetics of Antibiotic Interactions

The interactions of antibiotic molecules with bacterial cells and extra-cellular enzymes (β-lactamases) are governed by defined kinetic parameters inputted by the user. The interactions between antibiotics and β-lactamases in the environment conform to Michaelis-Menten kinetics. In this case, the key parameters that determine the reaction rate are the Michaelis constant (K_M), which is associated with the binding affinity, and the turnover rate of the reaction (k_{cat}). The ratio of these values (k_{cat}/K_M) is a convenient way of measuring enzyme efficiency (Zygmunt et al., 1992).

The interaction between antibiotic and PBP2a, on the

cell surface of the bacteria, is a pre-steady-state reaction (Lu et al., 1999). Therefore, the key parameters for describing this reaction are the dissociation constant (K_d) and the acylation rate of PBP2a (k_2). The ratio of these two values is defined as the second order rate constant of the reaction (k_2/K_d). This is a measure of the efficacy of the antibiotic at inhibiting PBP2a function.

Various laboratory techniques exist for calculating these kinetic parameters. Table 1 lists the kinetic parameters for the β-lactam antibiotics used in this study, derived from biological literature (Zygmunt et al., 1992; Lu et al., 1999; Fuda et al., 2004; Graves-Woodward & Pratt, 1998). These kinetic parameters were inputted into the model and their emergent properties analysed at the population-level.

Table 1: Key Biochemical / Kinetic Parameters of Some Common β-Lactam Antibiotics Versus Type A β-lactamase and PBP2a

Antibiotic	*Half-life (min)*	*β-lactamase k_{cat}/K_M $(M^{-1}s^{-1})$*	*PBP2a k_2/K_d $(M^{-1}s^{-1})$*
Penicillin G	42	3.3×10^6	16.5
Ampicillin	60	1.2×10^6	5
Cephalothin	30	2.2×10^3	1.96

Program Flow Structure

A generalized schema of the Micro-Gen program flow structure is depicted in Figure 2. The program cycles through a main loop, during which the various functions of the model are implemented, such as diffusion, bacterial activity and antibiotic interactions (Fig. 3). This loop represents one time-step of the simulation and is equivalent to 1.9 seconds of real-time. This corresponds to a simulated doubling time of 29 minutes for the MRSA cells under optimal conditions.

The real-time equivalent was calculated by running the simulator first with optimal growth conditions specified, and outputting a standard bacterial growth curve. The average simulated doubling time of the bacteria was calculated (in simulation time steps, or number of loops) during the exponential phase of growth. This result was then correlated with the optimal MRSA doubling time observed in experimental studies of 29 minutes, to determine the real-life equivalent time. Other model parameters are calibrated to the specified time-scale. Both the spatial and temporal resolutions of the model are fully configurable, by modifying the patch size and/or time-scale of the program loop.

The first step of the program loop (world evolution) is when Fick's law of diffusion is applied to the extra-

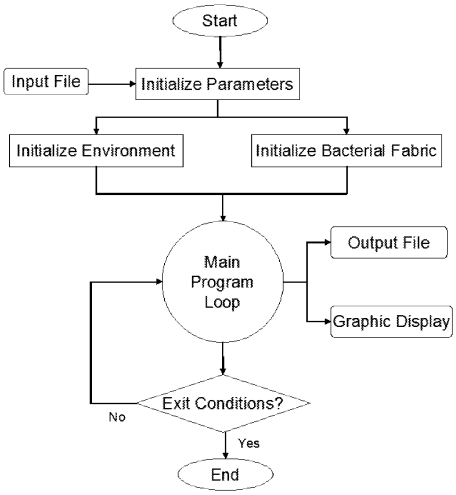


Figure 2: Schematic of Micro-Gen program flow structure.

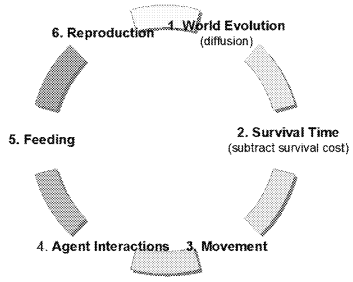


Figure 3: The Main Stages in the Micro-Gen Program Loop. Each Traversal of the Loop Represents One Time Step in the Simulation (1.9 s in Real-Time)

cellular molecules in the environment - antibiotics, β-lactamase enzymes and nutrients. The following step (survival time) is when the survival cost is subtracted from the energy reserves of the bacterial agents. This corresponds to energy expended in normal metabolic activities. If the bacterial energy level drops to zero at this stage the cell dies. During the movement stage, the cell shunting algorithm is applied if the patch is overcrowded. Otherwise, since *S. aureus* cells are non-motile, there is no active movement. The 'agent interactions' step is when the kinetic rules for interactions between Antibiotic-β-lactamase or Antibiotic-PBP2a are applied. Finally, the bacterial agents may feed and/or reproduce if nutrient is available or the internal energy reserve has surpassed a critical threshold.

Parallelisation

The original serial algorithm has been parallelised using MPI (Message Passing Interface), so that large simulations can be run on distributed computing clusters.

This is implemented using the strategy of domain decomposition, whereby the environment of the model is divided among multiple processors. The internal boundary conditions are shared between adjacent nodes, with quantities such as nutrient, antibiotic and enzyme levels, in boundary patches, passed between nodes every loop (Fig. 4).

Bacterial agents are passed between nodes by sending their constituent cell traits as sets of variables of the basic MPI types (MPI_INT etc.) and activating bacterial objects on the receiving nodes with these parameters. This is required since MPI has no in-built mechanisms for communicating software objects between nodes. The performance penalty of re-constructing bacterial agents on receiving nodes is minimized by the use of the bacterial 'fabric' data structure, which contains a reserve of pre-initialized bacterial objects.

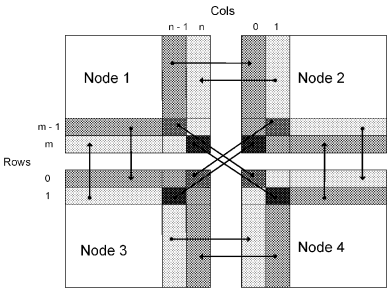


Figure 4: Illustration of Overlapping Boundary Conditions Between Adjacent Nodes in a 4-Node Parallel Computing Configuration

RESULTS

Some simulations were carried out to test the ability of Micro-Gen to predict the effects of three common antibiotics on the development of MRSA colonies. The kinetic parameters used to describe these antibiotics are summarized in Table 1. The importance of antibiotic half-life and β-lactamase production rate on treatment outcome was also investigated. This demonstrates the usefulness of the modelling approach for investigating the effects of individual parameters on the overall system dynamics.

Minimum Inhibitory Concentration

The Minimum Inhibitory Concentration (MIC) of antibiotic required to inhibit bacterial growth, was calculated from the model for a number of common antibiotics against MRSA, and compared with real-world results (Table 2). The MIC of an antibiotic is a standardized measure of the efficacy of the antibiotic against a particular strain of bacteria in culture. In order to convert the output from simulation units to real-world units the results were expressed relative to the MIC of Penicillin G (72.1 μg/ml). The predicted values from these initial tests match closely those recorded from *in vitro* experimental studies of MRSA (Zygmunt et al., 1992).

Table 2: Predicted MICs (Minimum Inhibitory Concentrations) Versus Experimentally Determined MICs (Zygmunt et al., 1992) of Various Antibiotics against MRSA Bacteria

Antibiotic	*Predicted M.I.C. ($\mu g/ml$)*	*Experimental M.I.C. ($\mu g/ml$)*
Penicillin G	72.1	72.1±7.5
Ampicillin	95	89.3±17.0
Cephalothin	103	91.3±8.2

Antibiotic Half-Life

In order to investigate the effect of antibiotic half-life on the outcome of treatment of MRSA, it was varied over the range 5-480 minutes for two common antibiotics Penicillin G and Cephalothin (Fig. 5). It can be seen that variation in the antibiotic half-life has a more pronounced effect on the efficacy (length of time growth is inhibited) of Cephalothin than Penicillin G. Increasing the half-life of Penicillin G by sixteen-fold from 30 minutes to 480 minutes results in $< 10\%$ increase in efficacy. On the other hand, for Cephalothin, there is nearly a ten-fold increase in efficacy over the same range.

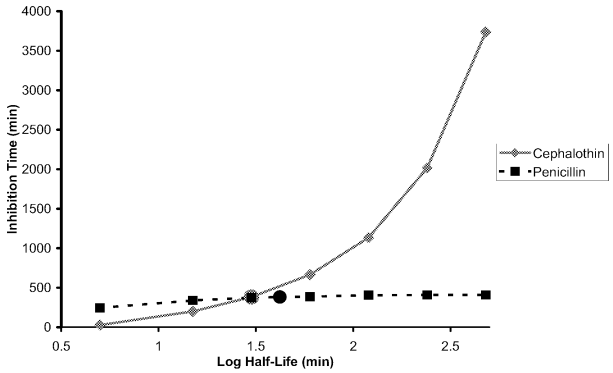


Figure 5: Effect of Varying Half-Life of Antibiotic on Inhibition Time (Antibiotic Dosage Equals MIC). Round Points Represent Natural Values.

β-Lactamase Production

The rate of β-lactamase production by MRSA cells was varied over a range of several orders of magnitude in order to compare the effects of β-lactamase levels on the inhibition times of these two antibiotics (Fig. 6). At low levels of β-lactamase production ($< 10^{-8}\,\mu$mol

s^{-1}) there is no correlation between production rate and inhibition time for Cephalothin (r = 0.0). By contrast, there is a strong negative correlation between β-lactamase production rate and inhibition time of Penicillin G over the entire range of values (r = - 0.958). This is because the enzyme is more catalytically efficient at cleaving Penicillin G molecules than Cephalothin (see Table 1).

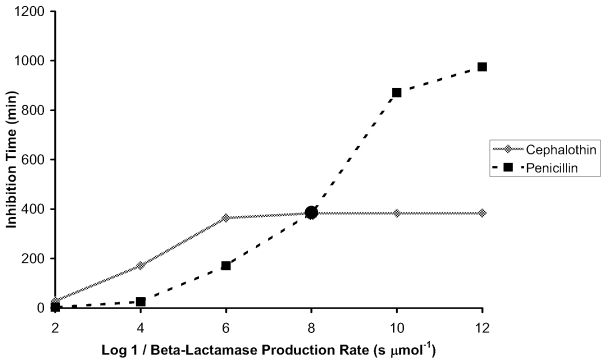


Figure 6: Effect of varying β-lactamase production rate on inhibition time (antibiotic dosage equals MIC). Round point represents natural value.

Parallel Efficiency

The parallelization of the model allows it to be scaled up in size and complexity to take advantage of high performance computing resources. Table 3 shows results from testing the parallel efficiency of the code on a 16 node computing cluster consisting of 16 x 2.8 GHz Intel Xeon Processors. The total size of the environment was 1200 x 1200 patches and the maximum size the bacterial culture reached was 6.8×10^6 bacterial agents, from a starting size of 16. The total memory requirement of the test simulation was 850 MB.

Table 3: Parallel Efficiency of Micro-Gen When Run on a 16 Node (Intel Xeon 2.8 GHz) Computing Cluster

No. of Nodes	*Parallel Efficiency (%)*
4	97.6
9	97.2
16	91.1

DISCUSSION

The antibiotics used for the tests recorded here were chosen because of the availability of detailed and reliable biochemical information on their structures and kinetics. Penicillin G and Cephalothin are members of two important classes of antibiotics: the Penicillins (which includes Methicillin and Ampicillin) and the Cephalosporins. The results are encouraging, but future work will include extending the tests to cover a broader range of antibiotics. Nevertheless, a number of insights can be made into the response of MRSA to these two types of antibiotic.

In order to validate the model for these antibiotics, the standard measure of antibiotic efficacy, the Minimum Inhibitory Concentration (MIC), was calculated and compared with real-world results (Table 2). The MIC is a high-level property that emerges from the dynamics of the system and it is often difficult to relate quantitatively knowledge at the molecular level with this standard measure of efficacy. However, by using Micro-Gen a clear relationship can be made between the MIC and the interactions of the basic cellular and molecular components of the system.

Some tests were carried out with the model to explore how individual parameters impact on the overall treatment outcome. The half-life of an antibiotic is an important determinant of treatment success. It can vary considerably in different environments, and it is often much reduced under *in vivo* conditions (i.e. in patients) compared to in laboratory conditions (*in vitro*). With the Micro-Gen model, the effect of changing the half-life of the antibiotic can be investigated allowing, for example, *in vitro* results to be related to the *in vivo* situation.

The results suggest that a reduction in half-life would have a more pronounced impact on treatment outcome with Cephalothin than with Penicillin G. This could have important clinical implications, since the relative efficacy of Cephalothin determined under laboratory conditions would not be an accurate indicator of its efficacy when administered to patients. Micro-Gen can be used to predict the relative reduction in efficacy under *in vivo* conditions and thus the treatment regimen could be modified to take this into account.

Another key determinant of bacterial survival in the presence of antibiotic is the production rate of β-lactamase enzyme by the cells. Again there is a marked difference in the reaction of the two classes of antibiotic to this parameter. The length of time growth is inhibited by Penicillin G is strongly influenced by the amount of β-lactamase produced. However, Cephalothin is only affected at very high β-lactamase concentrations.

This information can be used as a guide to determine the type of antibiotic most effective against a particular strain of MRSA. For example, they indicate that Penicillin G would be significantly more effective than Cephalothin against a β-lactamase-negative strain. However, by contrast, an MRSA strain with a high β-lactamase production rate would be more susceptible to Cephalothin. Since the β-lactamase production rates of

different strains of MRSA can vary by several orders of magnitude, it is important to be able to predict how treatment will be impacted by these changes.

The results presented here indicate that Micro-Gen can be used to successfully model resistance strategies in MRSA bacteria. It is highly adaptable and scalable from low performance personal computers up to high performance parallel computing environments. Since it uses an agent-based approach, the attributes of individual bacterial cells can be varied to explore the effects of spatial heterogeneity in a colony. This is its main advantage over mathematical models that assume a homogeneous environment with population-averaged parameters.

The ability to model the susceptibility of bacterial strains to different antibiotics based on low-level structural information about the antibiotic molecules is an important asset in rational drug design. The logistics involved in growing bacterial cultures and testing with novel candidate drug compounds requires a significant investment in time and resources. Micro-Gen can be used to facilitate the rapid screening of drug compounds against a wide array of bacterial strains in simulated culture conditions. In particular, the ability to predict the MIC for antibiotics is a significant, novel result as this is the primary determinant used to assess and compare the efficacies of different antibiotics in the clinical setting.

Future work will include integrating a genetic component in the bacterial agents. By incorporating this in the model, it would be possible to simulate the evolution of bacterial resistance in response to prolonged antibiotic exposure. The tool could also be used as a predictor of how combinations of antibiotics, or hypothetical antibiotics, would interact with bacterial cells.

Acknowledgements. The authors wish to acknowledge the contribution of Mathieu Joubert for developing a C++ version of the BAIT software tool.

AUTHOR BIOGRAPHY

James T. Murphy graduated from University College Dublin, Ireland, in 2002 with a Bachelor of Science degree in Cell & Molecular Biology. He completed a Master's degree in Bioinformatics at the School of Computing, Dublin City University in 2004. He is currently completing a Ph.D in the Modelling & Scientific Computing research group in D.C.U.

REFERENCES

Ender, M., McCallum, N., Adhikari, R., and Berger-Bachi, B. 2004. "Fitness cost of sccmec and methicillin resistance levels in staphylococcus aureus". *Antimicrobial Agents and Chemotherapy*, 48(6), 2295–2297.

Fisher, J. F., Meroueh, S. O., and Mobashery, S. 2005. "Bacterial resistance to beta-lactam antibiotics: compelling opportunism, compelling opportunity". *Chemical reviews*, 105(2), 395–424.

Fuda, C., Suvorov, M., Vakulenko, S. B., and Mobashery, S. 2004. "The basis for resistance to beta-lactam antibiotics by penicillin-binding protein 2a of methicillin-resistant staphylococcus aureus". *The Journal of biological chemistry*, 279(39), 40802–40806.

Fuda, C. C., Fisher, J. F., and Mobashery, S. 2005. "Beta-lactam resistance in staphylococcus aureus: the adaptive resistance of a plastic genome". *Cellular and molecular life sciences*, 62(22), 2617–2633.

Giesbrecht, P., Kersten, T., Maidhof, H., and Wecke, J. 1998. "Staphylococcal cell wall: morphogenesis and fatal variations in the presence of penicillin". *Microbiology and molecular biology reviews*, 62(4), 1371–1414.

Ginovart, M., Lopez, D., and Valls, J. 2002. "Indisim, an individual-based discrete simulation model to study bacterial cultures". *Journal of theoretical biology*, 214(2), 305–319.

Graves-Woodward, K. and Pratt, R. F. 1998. "Reaction of soluble penicillin-binding protein 2a of methicillin-resistant staphylococcus aureus with beta-lactams and acyclic substrates: kinetics in homogeneous solution". *The Biochemical journal*, 332(3), 755–761.

Jennings, N. R., Sycara, K., and Wooldridge, M. E. . 1998. "A roadmap of agent research and development". *Autonomous Agents and Multi-Agent Systems*, 1(1), 7–38.

Kotra, L. P. and Mobashery, S. 1998. "[beta]-lactam antibiotics, [beta]-lactamases and bacterial resistance". *Bulletin de l'Institut Pasteur*, 96(3), 139–150.

Lewis, R. A., Curnock, S. P., and Dyke, K. G. 1999. "Proteolytic cleavage of the repressor (blai) of beta-lactamase synthesis in staphylococcus aureus". *FEMS microbiology letters*, 178(2), 271–275.

Lu, W. P., Sun, Y., Bauer, M. D., Paule, S., Koenigs, P. M., and Kraft, W. G. 1999. "Penicillin-binding protein 2a from methicillin-resistant staphylococcus aureus: kinetic characterization of its interactions with beta-lactams using electrospray mass spectrometry". *Biochemistry*, 38(20), 6537–6546.

Murphy, J. T. and Walshe, R. 2006. "Micro-gen: An agent-based model of bacteria-antibiotic interactions in batch culture". In *Proceedings of the 20th Annual European Simulation and Modelling Conference* (pp. 239–242).: Eurosis-ETI.

Walshe, R. 2006. "Modelling bacterial growth patterns in the presence of antibiotic". In *Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems* (pp. 177–188). Washington, DC, USA: IEEE Computer Society.

Zygmunt, D. J., Stratton, C. W., and Kernodle, D. S. 1992. "Characterization of four beta-lactamases produced by staphylococcus aureus". *Antimicrobial Agents and Chemotherapy*, 36(2), 440–445.

Adaptative Dichotomic Optimization: a New Method for the Calibration of Agent-Based Models

Benoît CALVEZ and Guillaume HUTZLER
Universite d'Evry-Val d'Essonne/CNRS
IBISC, FRE 2873
523, Place des terrasses de l'Agora 91000 Evry, France
{benoit.calvez,guillaume.hutzler}@ibisc.univ-evry.fr

KEYWORDS
Model design, Parameter identification, Optimization, Discrete simulation, Object-oriented

ABSTRACT

In this paper, we propose a new approach for the exploration of the parameter space of agent-based models: Adaptative Dichotomic Optimization. Agent-based models are generally characterized by a great number of parameters, a lot of which cannot be evaluated with the current knowledge about the real system. The aim of the work is to provide tools for the calibration of these models, which consists in finding the optimal set of parameters for a given criterion. The criterion can be for example that the model achieves a specific function optimally or that the results of the simulation are as close to possible of experimental data. Our approach is based on the partition of the parameter space (the interval of variation of each variable is divided into a finite number of intervals) and on a parallel exploration of the various parameters by the agents of the model. The navigation in the parameter space is done by grouping or dividing adaptively some of the intervals, according to an algorithm which is adapted from Ant Colony Systems.

INTRODUCTION

Agent-based simulation (ABS) is interested in the modelling and the simulation of complex systems. Its aim is to reproduce the dynamics of real systems by modelling the entities as agents, whose behavior and interactions are defined, and which "live" in a simulated environment. A first validation of such models is obtained by comparing the resulting dynamics, when the model is simulated, with that of the real system (measured thanks to experimental data).

One of the crucial aspects of the design process lies in the tuning of the model. Indeed, this kind of model is generally characterized by lots of parameters which together determine the global dynamics of the system. The search space is thus gigantic. Moreover, the behavior of these complex systems is often chaotic: on the one hand small changes made to a single parameter sometimes lead to a radical modification of the dynamics of the whole system; on the other hand some emergent phenomena are only produced in very specific conditions and won't occur if these conditions are not met. The solution space can thus be very small. As a consequence, the development and the parameter setting of an agent-based model may become long and tedious if we have no accurate, automatic and systematic strategy to explore the parameter space.

The development of such a strategy is precisely the object of this paper, which follows several works on the subject (see next section). All of these, however, appeared to be too limited, either because they achieved only a limited exploration of the parameter space, or because they required a great number of simulation runs in order to converge. The main difference between classical optimization and the calibration of agent-based models is but that the evaluation of a single set of parameters generally requires at least a complete simulation run, which may take quite a long time. If the model is stochastic, it may even require several runs for the same parameter set, in order to have a good confidence in the evaluation. It is thus most important that the number of evaluations be reduced to its minimum if the method is to be usable.

The key idea of our approach is to take advantage of the natural concurrency in agent-based models, so as to achieve a parallel exploration of the parameter space. In this approach, which we call Adaptive Dichotomic Optimization (ADO), every individual agent is instantiated with its own specific parameter values, chosen in a finite number of intervals, for each variable. The different agents of the model thus have completely different settings. The basic intuition is that the "performance" of the resulting model will be statistically better if it includes a larger proportion of agents with "good" parameters. The intervals in which the parameters have been chosen for the agents of a given model will thus be rewarded depending on the performance of the model. The exploration of the parameter space will then be refined adaptively by merging together adjacent intervals with low rewards and by dividing in two intervals with high rewards.

The next section describes previous works on the subject and discusses their respective strengths and weaknesses. We then present our approach in more details. After we discuss our results before concluding.

PREVIOUS WORKS

Different methods have already been proposed to explore automatically the parameter space of discrete models. In the `NetLogo` platform for instance, the "BehaviorSpace"

(Wilensky 1999) tool allows to explore automatically and systematically the parameter space. This space is a Cartesian product of values that each parameter can take, some of them being chosen as fixed, others being limited to a subset of all possible values. However when the model has lots of parameters, some of which can take a good many values (real-valued parameters for example), the parameter space becomes huge and the systematic exploration becomes impossible. The number of values tested for each parameter will thus be very low and the corresponding exploration limited to a very small part of the whole parameter space.

Other methods have been proposed, which differentially explore the whole parameter space, focusing on the most interesting areas. That's the case of the method developed by Brueckner and Parunak (Brueckner and Parunak 2003). They use a "parameter sweep infrastructure", which is similar to the "BehaviorSpace" tool of `NetLogo` . However, to avoid a systematic exploration, they use searcher agents and introduce the notion of fitness. The aim of a searcher agent is to travel in the parameter space to look for the highest fitness. Starting from a given location in the parameter space, searcher agents have two choices: move or simulate. Each agent chooses according to the confidence of the fitness estimate (proportional to the number of simulations at this point) and the value of the fitness. If it chooses to move, it heads for the neighboring region with highest fitness. A disadvantage of this method is that searcher agents may head for local fitness maxima.

Another method is to add knowledge to the agent-based model, as is the case with white box calibration (Fehler et al. 2004). The principle is to use the knowledge of the agent-based model to improve the tuning process. The aim is to reduce the parameter space by breaking down the model into smaller submodels, which can be done using different methods (General Model Decomposition, Functional Decomposition, ...). Each of the submodels is then calibrated, before merging them back to form the model. The division and fusion operations are the difficult steps of the method. The division operation, on the one hand, requires the addition of knowledge about the model, which may not be available. The fusion operation, on the other hand, has to merge calibrated submodels into a calibrated higher model, which is not automatic.

Another approach is to consider the problem of the development and the validation of agent-based models as an optimization problem. The validation can thus be reformulated as the identification of a parameter set that optimizes some function. The optimization function would be for example the distance between the artificial model that we simulate and the real system. To solve this optimization problem, several authors (Calvez and Hutzler 2005, Rogers and von Tessin 2004, Narzisi et al. 2006) propose to use genetic algorithms with different variants.

The use of these method needs the computation of the fitness function. The trouble, as we underlined in the introduction, is that the computation of the fitness function for a single model potentially requires several simulation runs, each of which may last for several hours. Without a powerful computational cluster, the method is therefore difficult to apply.

ADO GLOBAL VISION

The basic idea is to take advantage of the fact that agent-based simulations rely on multiple agents. Instead of considering that all of them should be parameterized in the same way, we propose to enable the parameterization of the different agents with different settings. Instead of evaluating parameter settings one at a time, we can thus evaluate several settings in parallel in a single model. The hypothesis that we made is that if such a model behaves "correctly", then the agents that compose the model are considered to have adequate parameters. In a way, the idea is similar to the principle of Ant Colony Systems (Dorigo et al. 1999).

The other idea is to explore the parameter space differentially, depending on the potential interest of the different regions of the space. Taking inspiration from dichotomic search and from octrees, we consider that a parameter space of dimension n (n independent parameters) is initially divided into hypercubes of dimension n. Practically, the definition set of each of the parameters is initially divided into a fixed number of identical intervals. Then, for each individual parameter, depending on rewards received by the different intervals, we may differentially merge or divide the intervals. If an interval did receive high rewards, this signals an interesting area of the parameter space. The corresponding interval may thus be divided into two sub-intervals so as to have a more precise evaluation of the interesting parameters. On the contrary, if two adjacent intervals received few rewards, this signals an area of the parameter space of low interest. The corresponding intervals may thus be merged so as to stop waisting time at exploring this area.

When an agent is instantiated, the value of each parameter is chosen randomly among the intervals that divide the definition set of the parameter. After a model has been evaluated (by running a simulation and computing the fitness of the model), the intervals in which the parameters of the agents have been chosen are rewarded. For each parameter, and for each interval, the reward is proportional to the global fitness of the model and to the number of agents that have taken their parameter's value in the interval.

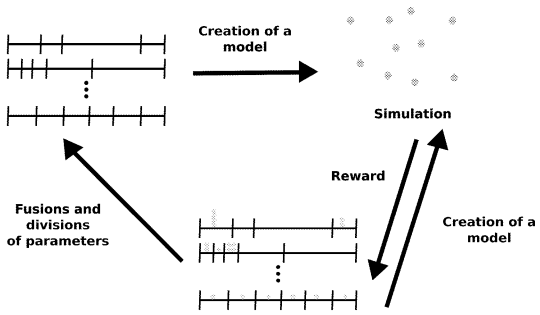


Figure 1: ADO Method outline

The global process of the Adaptive Dichotomic Optimization method proceeds as follows (see figure 1): a model is created, its fitness is computed, and the corresponding intervals are rewarded. This is iterated several times (it can also automatically be distributed on several computers), until the different intervals have received enough rewards so that it becomes significant. For each parameter independently of the others, the best interval (the one having the largest reward) is selected and divided in two. The whole process is itself iterated until it stabilizes.

PARAMETERS

In this method, we have made the working hypothesis that each parameter is independent of the other ones, that is to say the application of the ADO algorithm on one parameter can be performed independently of the other ones. In this case, our parameter space is broken down into n parameters divided into m_i intervals (the size of the search space is thus about $n \times \mathbb{R}$). In reality, the parameter space is a space of n dimensions (the size of the search space is about $\mathbb{R}^n$.). Without this hypothesis, in the selection phase, the number of intervals would raise by $2^n - 1$, which would cause serious difficulties. With this hypothesis, the number of intervals only raise by n.
This hypothesis can be justified by the fact that we run lots of simulations with lots of agents, which implies that the parameter space is globally covered. If we consider a single parameter, it has been evaluated with the other parameters taking a great number of different configurations. Thus, if an interval has received high rewards, we can be confident that this interval is interesting, disregarding the specific values of the other parameters. This hypothesis has also been confirmed experimentally: we ran several tests with different models, and it appeared to remain correct for all the configurations tested.

METHOD

We have developed several variants of the ADO method. In this paper we present one best variant.
We took inspiration from the Ant Colony Optimization approach(Blum 2005, Dorigo and Di Caro 1999) in order to improve our method for creating this variant : we made a parallel between pheromones in the Ant Colony System(Dorigo et al. 1996) model and the rewards given to the intervals in the ADO approach. More precisely, it is the mean reward which is chosen as pheromones.
Moreover we add a heuristic information and a confidence in this heuristic information. For each interval, we compute the heuristic information in three steps. In the first step, we create a model where all the agents are initialized with the same parameter set, which is chosen in the following way:

- for the specific interval, whose heuristic value we wish to estimate, we choose the median value of the interval
- for all the other parameters, we choose the median value of the parameter

In the second step, we simulate this model several times (because of the stochasticity). In the last step, we calculate the heuristic information: the value of the heuristic information is the average value of the fitnesses for the various simulation runs. The confidence in the heuristic information of an interval is 1 after its calculation. During a fusion stage, the confidence is the average of the two confidences of the two intervals. During a division step, the confidence is divided in two. We compute again the heuristic information when the confidence is lower than a given value.
The phase which the values of the parameter are chosen is modified to distinguish two steps. First, we choose an interval j of the parameter i with a probability proportional to

$$[\tau_{ij}]^{\alpha} \times [\eta_{ij}]^{\beta}$$

where τ is the pheromone quantity of this interval (that is to say the average reward), η is the heuristic information, and α and β are parameters that control the relative importance of the pheromone versus the heuristic information. Second, we choose a value in this interval with an uniform probability.
The division and fusion stage is also modified. An interval j of the parameter i is divided if $\tau_{ij} > \overline{\tau_i} + 2 \times \sigma_i$, where σ_i is the standard deviation. An interval j of the parameter i is merged if $\tau_{ij} < \overline{\tau_i} - \sigma_i$.

RESULTS

Model

We have tested our method on different models. In this paper, we will focus on a single model to illustrate and to test the method. To this end, we decided to use the well-known ant foraging model, provided by the modeling environment platform `NetLogo` (Wilensky 1998). There are several parameters in this model. We are interessed in four local parameters : `speed` (this parameter characterizes the speed of an agent. It varies between 0 and 20 patches per simulation step.), `patch_ahead` (this parameter characterizes the number of patches looked ahead to "sniff" the chemical. It varies between 0 and 10 patches.), `angle_vision` (this parameter characterizes the angle of vision. It varies between 0°and 360°.), `drop_size` (this parameter characterizes the initial quantity of chemical that the agents drop in the environment when they come back to the nest with items of food. It varies between between 0 and 200.).

Results

In our example, the model have 25 ants, we simulate 500 steps. The global parameters remain constant. We try to optimize the four local parameters. The fitness is the quantity of food brought back to the nest. To compare our method with previous works, we also ran a genetic algorithm method (described in the paper (Calvez and Hutzler 2005)). We stop

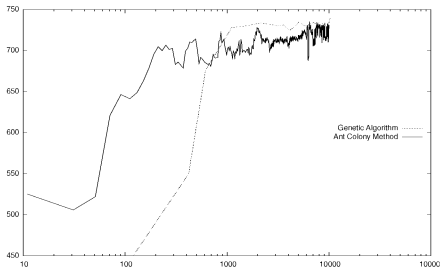


Figure 2: Evolution of the fitness according to the number of simulation runs

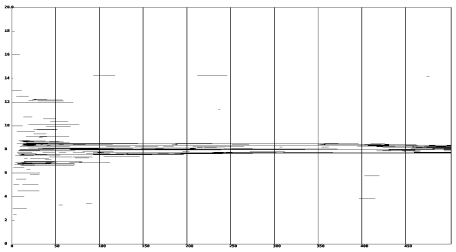


Figure 3: Evolution of divisions of parameter "speed"

	genetic algorithm method	ant colony method
speed	7.35	7.69
patch_ahead	9.76	9.84
angle_vision	193.76	195.18
drop_size	109.89	144.99

Table 1: Optimized parameters values

the method after 10000 simulations. Figure 2 shows the evolution of the fitness according to the number of simulation runs.

The interesting result is that with few simulation runs, the Ant colony version of the ADO method have better results than the GA method. In this example, with only 300 simulation runs, we have already a good approximation of the solution.

Moreover we have more information with the ADO method than with the genetic algorithm method. Figure 3 presents the evolution of the division of the first parameter in our ADO method. At the beginning of the algorithm, the parameter is divided into ten equal parts, then as the method runs, the divisions of the parameter evolve: we have a convergence of the divisions to accurate zones of parameters. This illustrates one of the interests of this method: we start with the entire parameter space before focusing on more interesting part but without forgetting though the other parts. Instead of providing only an optimal value, the method thus provides a kind of a cartography of the parameter space.

We can then study the solution parameters obtained by the two methods. Table 1 shows the optimized parameters. Whatever the method, the solutions have the same characteristics: the agents move with the big speed about 7.5, with a vision angle about 190°, with a farthest vision. The initial quantity of chemical is very high but without maximizing the quantity. To summarize, the best agent is the one that moves quick, and that looks ahead farthest, in front of it, which appears to be pretty logical.

DISCUSSION

We presented a basic algorithm and one variant, and we applied them on a simple benchmark, namely the ant foraging problem. The first results appear to be rather promising as they show a quick convergence of the ADO method towards a solution that is almost as good as the one that was found with genetic algorithms. Both methods are easily distributable on several computers so as to speed up the process.

A little drawback of the ADO approach is that is works best for agents' parameters, not so well for global parameters. However, this should not be too problematic since agents' parameters are generally the ones that are to be estimated, global parameters being estimated from experimental data.

As compared to genetic algorithms, we can but see two major advantages of the ADO approach. The first one is that convergence is very fast with few simulation runs, whereas a great number of runs is necessary to bootstrap the genetic algorithm. We can thus achieve a first exploration of the parameter space with only 100 or 200 simulation runs, which is potentially interesting for the modeller during the prototyping phase. The second advantage is that with the ADO method, the parameter space can, at any moment of the algorithm, be interpreted as a kind of a cartography of the parameter space, sparse intervals corresponding to low interest regions, dense intervals corresponding to high interest regions. Specific visualisation techniques remain to be developed so

as to take advantage of these characteristics.

CONCLUSION AND PERSPECTIVE

In this paper, we presented a new algorithm for the automatic calibration of agent-based models We also presented preliminary results on a simple benchmark with four continuous independent parameters to tune. These results suggest that the method is interesting especially when the execution of simulation runs is very costly, because of its very fast convergence. The argument is that agent-based models present characteristics that make their calibration difficult, mainly because of the long time required to execute simulation runs, as compared to the fitness functions usually used in genetic algorithms. Another characteristics of agent-based models is that they are concurrent models, property that we exploited so as to propose an efficient algorithm, inspired by the principles of dichotomic search and of ant colony systems. In addition, we underlined that the method enables an exhaustive cartography of the parameter space, which is a real advantage against genetic algorithms.

Further investigations still need to be undertaken to better understand the role of the number of agents in the optimization process. We will also study the signification of the patterns of division for the parameters and develop visualization tools in order to analyze these patterns. If a parameter shows a very densely divided area with very small intervals, this parameter is probably an important parameter in the dynamics of the simulation. On the contrary if the division of a parameter is relatively homogeneous, this parameter may not be essential for the model. Finally, we investigate the possibility to mix the different approaches, namely Adaptive Dichotomic Optimization and Genetic Algorithms, so as to combine the strengths and advantages of both methods.

In addition, we need to enlarge the set of models on which the method is evaluated, with larger parameter spaces. We are now on the process of applying the method to "real" models, that is models with true modelling questions, experimental data and hypothesis and theories. This will be the real test, that is, can our method give automatic answers to some of the modelling questions that are raised by the modeller. This is the real challenge.

REFERENCES

Blum C., 2005. *Ant colony optimization: Introduction and recent trends. Physics of Life Reviews*, 2, no. 4, 353–373.

Brueckner S.A. and Parunak H.V.D., 2003. *Resource-aware exploration of the emergent dynamics of simulated systems.* In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM Press, New York, NY, USA. ISBN 1-58113-683-8, 781–788.

Calvez B. and Hutzler G., 2005. *Automatic tuning of agent-based models using genetic algorithms.* In L. Antunes and J.S. Sichman (Eds.), *Proceedings of the 6th International Workshop on Multi-Agent Based Simulation (MABS'05).* Luis Antunes, Jaime Simao Sichman, *lnai*, vol. 3891, 39–50.

Dorigo M. and Di Caro G., 1999. *The Ant Colony Optimization Meta-Heuristic*, McGraw-Hill, chap. New Ideas in Optimization. 11–32.

Dorigo M.; Maniezzo V.; and Colorni A., 1996. *The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26, no. 1, 29–41.

Dorigo M.; Maniezzo V.; and Colorni A., 1999. *Positive Feedback as a Search Strategy. Technical Report No 91-016, Politecnico di Milano, Italy, 1991.*

Fehler M.; Klügl F.; and Puppe F., 2004. *Techniques for Analysis and Calibration of Multi-agent Simulations.* In M.P. Gleizes; A. Omicini; and F. Zambonelli (Eds.), *ESAW*. Springer, *Lecture Notes in Computer Science*, vol. 3451. ISBN 3-540-27330-1, 305–321.

Narzisi G.; Mysore V.; and Bud Mishra B., 2006. *Multi-Objective Evolutionary Optimization of Agent-based Models: An Application to Emergency Response Planning.* In B. Kovalerchuk (Ed.), *The IASTED International Conference on Computational Intelligence (CI 2006).*

Rogers A. and von Tessin P., 2004. *Multi-Objective Calibration For Agent-Based Models.* In *5th Worksho on Agent-Based Simulation.*

Wilensky U., 1998. *NetLogo Ants model.* Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky U., 1999. *NetLogo*. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

MULTI-AGENT SYSTEMS AND SIMULATION

Foreword

It is our great pleasure to welcome you to the 2nd edition of the MAS&S (Multi-Agent Systems and Simulation) workshop that takes place this year at the ESM 2007 Conference, in S. Julians, Malta.

Multi-agent systems (MASs) provide powerful models for representing real-world environments with an appropriate degree of complexity and dynamism. Several industrial and academic experiences have already shown that the use of MAS offers advantages in manufacturing processes, e-Commerce, network management, and, more generally, complex systems analysis. Since MASs in such contexts are to be tested before their actual deployment and execution, methodologies that support validation through simulation (e.g. discrete-event simulation) of the MAS under development are highly required. In fact, simulation of a MAS can not only demonstrate that a MAS correctly behaves according to its specifications but can also support the analysis of emergent properties of the under-test MAS.

In this workshop we are interested in the combined issues of multi-agent systems and simulation particularly in the context of Agent Oriented Software Engineering (AOSE).

The areas of interest of the workshop are the following: Agent-based simulation techniques and methodologies; Discrete-event simulation of Multi-Agent Systems; Simulation as Validation Tool for the development process of MAS; Agent-oriented Methodologies incorporating Simulation tools; MAS Simulation driven by formal models; MAS Simulation toolkits and frameworks; Testing vs. Simulation of MAS; Agents as modeling concept enabling communication between domain experts and simulationists; Industrial case studies based on MAS and simulation/testing; Agent Computational Economics/Agent Computational Finance.

The ten papers selected for presentation offer a very interesting and broad range of views on methodologies, models, architectures, and applications integrating Multi-Agent systems and Simulation.

The quality and variety of the topics covered by the presented papers reflect the vitality of this research area. The work of the program committee members guaranteed an accurate review process, with at least two reviews per paper, and assured a selection of good quality papers that reflect current research efforts and are a valuable source of information for the researchers that want to become familiar with the field of Multi-Agent Systems and Simulation.

We hope that you will find this program interesting and thought-provoking and that the workshop will provide you with a valuable opportunity to share ideas with other researchers and practitioners from institutions around the world.

Massimo Cossentino
ICAR-CNR, Italy

Giancarlo Fortino
University of Calabria, Italy

Juan Pavon
Universidad Complutense Madrid, Spain

AGENT ORIENTED METHODOLOGIES

MULTI-AGENT SYSTEMS FORMALIZATION USING FORMAL METHOD DPLA

Henrikas Pranevicius
Dalius Makackas
Agne Paulauskaite-Taraseviciene

Department of Business Informatics, Faculty of Informatics
Kaunas University of Technology
Studentu 56, Kaunas, Lithuania
E-mail: hepran@vik.ktu.lt, damaka@vik.ktu.lt, agne@ifko.ktu.lt

KEYWORDS

Multi-agent systems, formalization, dynamic structure, PLA, DEVS.

ABSTRACT

Since the number of multi-agent systems grows and such systems are usually large-scale and complex, it is useful to use formal methods for their specification and validation. Besides their specific features, multi-agent systems are primarily dynamic systems with capabilities to change not only their state in time, but all the structure as well. Many formal methods are able to describe static structure systems. A piece-linear aggregate (PLA) is timed automata based specification formalism, which is widely used for creating simulation models and their validation. In this paper we present the extended PLA formalism – DPLA, which is able to specify dynamic structure systems including agent-oriented systems as well. Realization of DPLA of a concrete multi-agent application is presented. Using such a realization agents capabilities for reaching their goals are tested.

INTRODUCTION

Agents and their systems are a relatively new research trend, but recently we can see more publications in which agents and multi-agents systems (MAS) are used for solving various problems. These systems are widely applied in information, control, commercial systems, etc. Development of an agent-based system corresponds to a system, which could be able to perform the selected tasks in a certain environment. Like in modeling all computer systems, it is important to define correct, unambiguous concepts of agents and their properties, actions, interactions with each other and the environment. The cost is also a very significant aspect in multi-agent systems design. In this case it is useful to use the virtual environment, which appliance reduces costs and which allows to test the behavior of a system in various situations. Since MAS are often used for large-scale and complex systems, there should be possibilities of system analysis, validation and verification.

Specifying multi-agent systems using formal methods enables the unambiguous description of all concepts and behavior of MAS. The opportunity to check MAS specification correctness and consistency is the main advantage of formal methods.

A piece-linear aggregate (PLA) is timed automata based specification formalism, which is widely used for creating simulation models (for simulation of computer network protocols often) and their validation; also it is very important for designing complex real time systems. For formalization MAS PLA formalism was extended to DPLA and is presented in this paper. The usage of such DPLA provides a possibility to specify all dynamic features of such systems.

MAS FORMALIZATION POSSIBILITIES

Conceptually, multi-agent systems can be considered as dynamic structure systems with their specific features. Multi-agent systems interact in a dynamic environment changing the type and number of its objects, where objects can change their structure in time and perform autonomously (Wooldridge 1999; Fox et al. 2003). Whereas such systems are mostly large and complex, MAS comes into the existence for motivation of the formal specification. Most of formal methods enable the specification of static structure systems, which mean that dynamic features of MAS are complex to describe or can't be described at all. For expanding the range of specified systems, formal notations should be extended with capabilities to describe all features of such systems. For example, to describe dynamic structure systems, the classical notation of DEVS (discrete event system specification) was extended by adding special functions (Uhrmacher 2001a). In dynDEVS structural changes are invoked by a model transition function ρ_α, which generates "incarnations" of dynamic DEVS and a network transition function ρ_N, which keeps the state and structure of models that belongs to the composition of network. Using the dynamic DEVS notation the existing models can be deleted or new created, couplings between models added and removed. Such capabilities of dynDEVS are also useful for modeling MAS (Rohl and Uhrmacher 2004; Uhrmacher 2001b). For various experiments, the system JAMES (A Java-Based Agent Modeling Environment for Simulation) was created (Schattenberg and Uhrmacher 2001) in which agents are modeled as time triggered state automata. The virtual environment allows creating different scenarios and testing the behavior of agents.

Agent systems are also modeled with different Petri nets (e.g. mobile Nets, Colored Petri Nets, high level Petri nets). For example, modeling MAS with high level Petri nets "individual agents are modeled as tokens so that they migrate from one component to another by transition firing at runtime" (Xu and Deng 2000). Colored Petri nets are mostly used for modeling and testing the social behavior of agents (Weyns and Holvoet 2002).

We can find publications in which MAS are mentioned and described in Z specification language (d'Inverno et al. 2004). The authors of these publications emphasize that there is no common terminology for agents. Since "the term of agent is now used so frequently that there is no commonly accepted notation of what it is that constitutes an agent" (Luck and d'Inverno 2003), they decided to describe MAS as an environment, which have attributes. Objects are considered as collections of attributes, agents as objects with goals and autonomous agents as agents with self-motivation. Their main goal is to have clear formal definitions of various agents, irrespective to dynamics of MAS. However, the description of some specifications is difficult to understand. Furthermore, a lack of tools for automated code generation into object-oriented languages exists.

It is difficult to determine which methods are better than others, because all these formal methods specifying MAS highlight and test different features of agents. As a result of the overview, we can make a consumption that automaton based formal methods are more suitable for describing and testing multi-agent systems. That is the reason for deciding to use the formal method PLA for MAS formalization.

DYNAMIC APPROACH OF PLA

A piece-linear aggregate (PLA) is timed automata based specification formalism, which is widely used for creating simulation models (for simulation of computer network protocols often) and their validation (Pranevicius 2003). In the application of the aggregate approach a system is represented as a set of interacting piece – linear aggregates. The PLA is taken as an object defined by a set of states Z, input signals X, and output signals Y. The aggregate functioning is considered in a set of time moments $t \in T$. The state $z \in Z$, input signals $x \in X$, and output signals $y \in Y$ are considered to be time functions. Apart from these sets, transition H and output G operators must be known as well.

DPLA: Dynamic PLA

The dynamic structure is an important capability, related to the hierarchical structure, for simulating complex systems. PLA is very useful for designing complex real time systems, but only for static structure systems. In multi-agent systems the environment is varying in time changing the number of its attributes (e.g. new agents are added or removed). Agents can change their internal features, besides due to environmental changes they can migrate, create and remove other agents. Using PLA, agents and environment can be considered as aggregates. Such agent-oriented features as goals, believes and desires are discrete components of an aggregate that represents an agent. The interaction between agents and environment can be easily represented through couplings (channels for input and output signals). Controlling sequences can denote the duration of actions of agents (e.g. deliberation time). Transitions operators define the logic of all actions. However, it is difficult to specify such essential characteristics as mobility or autonomy using the PLA notation. For specifying multi-agent systems we modify some components of the original PLA and add some extra operators. The difference between PLA and DPLA (dynamic PLA) is presented bellow in figure 1.

PLA	DPLA
Input signals $X = \{x_1, x_2, ..., x_N\}$	Input signals $X(t_m) = \{x_1, x_2, ..., x_N\}$
Output signals $Y = \{y_1, y_2, ..., y_N\}$	Output signals $Y(t_m) = \{y_1, y_2, ..., y_N\}$
External events $E' = \{e_1', e_2', ..., e_N'\}$	External events $E'(t_m) = \{e_1', e_2', ..., e_N'\}$
Internal events $E'' = \{e_1'', e_2'', ..., e_f''\}$	Internal events $E''(t_m) = \{e_1'', e_2'', ..., e_f''\}$
Discrete component $\nu(t_m) = \{\nu_1(t_m), \nu_1(t_m) ..., \nu_p(t_m)\}$	Discrete component $\nu(t_m) = \{A(t_m), R(t_m) ..., \nu_p(t_m)\}$
Number of system aggregates K	Set of children aggregates $A(t_m) = (Ag_name)$
Matrix of input signals $R = \{r_{ji}\}, i = \overline{1,L}, j = \overline{1,2}$	Set of relationships $R(t_m) = (name: source_Ag \mapsto t\arg et_Ag)$
Matrix of output signals $H = \{h_{ij}\}, i = \overline{1,K}, j = \overline{1, \max_{1 \le i \le K}\{M_K\}}$	$R: Y^* \to X^*, where$ $Y^* = \bigcup_{A_i \in A(t_m)} Y_{A_i}, \quad X^* = \bigcup_{A_j \in A(t_m)} X_{A_j}$
Continuous component $z_v(t_m) = \{w(e_f'', t_m)\}$	Continuous component $z_v(t_m) = \{w(e_f'', t_m)\}$
Controlling sequences $e_i'' \mapsto \{\xi_j^{(i)}\}$	Controlling sequences $V(t_m) = \{e_i'' \mapsto \{\xi_j^{(i)}\}\}$
Initial state $z(t_m)$	Initial state $z(t_m)$
Transition operator $H[z(t_m, e_i)]$	Transition operator $H[z(t_m, e_i)]$
Output operator $G[z(t_m, e_i)]$	Output operator $G[z(t_m, e_i)]$

Figure 1: Components of PLA and DPLA

Differently from original PLA, in DPLA the number of events, input, output signals, controlling sequences, discrete and continuous component may vary in time. Each aggregate can have children aggregates in its content. dynDEVS notation is similar to DPLA, but differently from dynDEVS coupled model (which is like a frame of its component), a parent aggregate is an ordinary aggregate with its own internal events, external event, outputs, inputs and so on.

In DPLA there are two types of dynamic structural changes:

1. The level of a single aggregate: Structural changes of a single aggregate (i.e. a new initial state can be added $E''(t_{m+1}) = E''(t_m) \cup e_i''$ or deleted $E''(t_{m+1}) = E''(t_m) / e_i''$).

2. The level of aggregate system: Structural changes between two or more aggregates (i.e., a new link between aggregates can be added, or an existing link deleted; also new aggregates can be created or eliminated).

Simulation Algorithm

The simulation process for aggregates in DPLA is almost the same as in classical PLA (Pranevicius 2003). Processing is event-driven, which is faster and more efficient than time-stepped simulation. The modeling algorithm of DPLA is described below:

Step 1. First of all initial states of all aggregates are defined: $z_k(t_0) = \{v_k(t_0); r_k(t_0); z_v^k(t_0)\}$, $k = \overline{1,K}$, where K is a number of aggregates, v_k – a discrete component of k aggregate, r_k - a number of events in the interval $[t_0,t_m]$.

Step 2. Using the equation (1) bellow, the next time moment of the internal event and the name of the aggregate in which that event will occur are determined.

$$t(_{m+1}) = \min_{1\le k\le K}\left(\min_{\substack{w_k(e_r'',t_m)\in z_{v_k}(t_m)\\ e_r''\in E_k''}} w_k(e_r'',t_m)\right) \qquad (1)$$

Step 3. A new state $z_k(t_{m+1})$ of k aggregate is defined by $z_k(t_{m+1}) = H[z_k(t_m), e_i]$, $e_i \in E''$ and the output signal is generated $y = G[z_k(t_m), e_i]$, $e_i \in E'', y \in \widetilde{Y}$. New states of the rest of aggregates are recalculated $z_i(t_{m+1}) = z_i(t_m)$ $i = \overline{1,K}$, $i \neq K$. Continuous and discrete components of aggregates, which number is not equal to k, are unchanged.

Step 4. The set $\widetilde{Y}$ is checked. If there is at least one element $y_i \in Y_k$, the name of the target aggregate, which receives this signal y_i is determined, using a set of relationships R (go to the step 5). If $\widetilde{Y} = \varnothing$ return to step 2.

Step 5. A new state of the target aggregate is defined after it receives an input signal and an output signal is generated, which is traced in $\widetilde{Y}$. The element y_i is deleted from $\widetilde{Y}$ (go to the step 4).

For dynamics in DPLA, we create four protocols for defining the sequence of actions of structural changes: the creation of relationships (CR), the deletion of relationships (DR), the creation of new aggregates (CA), and the deletion of aggregates (DA). Performing structural changes by these protocols, an aggregate can evolve in the process and differently from dynDEVS, we don't have to predefine all possible models variations of each aggregate (Uhrmacher 2001).

The sequence of actions of the CR protocol is presented above (cp. Figure 2). If one aggregate (Ag1), which belongs to the coordinator, decided to create a link with another aggregate (Ag2) belonging to the same coordinator, it sends a message "CR" to its parent coordinator. The message "CR" keeps all the information needed for creating a new link (a name of link, associated aggregates). The coordinator checks, if such aggregates exist in its set of children aggregates ($Ag1, Ag2 \in A(t_m)$). If there is none of these aggregates, the coordinator sends the "CR_error" message to the aggregate, which initiates "CR". Otherwise, the coordinator forwards the "Do_CR" message to additional aggregates. When the aggregate gets the message "Do_CR", it is preparing for such actions: if a signal with my_ID=target_Ag exists in the content of CR, a new signal has to be added in its input signals set $X(t_{m+1}) = X(t_m) \cup x_{name}$, an additional external event as well $E'(t_{m+1}) = E'(t_m) \cup e'_{name}$. The transition operator for a new external event is activated $H(e'_{name})$. After these changes it sends a "prepared_Do_CR" message back to the parent coordinator. Meantime, the coordinator is waiting for "prepared_Do_CR" from all expected aggregates. Whenever it gets all answers, structural changes in its level start. The coordinator creates links, includes them to its relationship set $R(t_m)$ and sends the message "Done_CR" for additional children aggregates (Ag1, Ag2). When the aggregate gets the "Done_CR" message, it starts checking if there exists a signal with my_ID=source_Ag in the content of "Done_CR". The output signal is included in the set of output signals $Y(t_{m+1}) = Y(t_m) \cup y_{name}$ and an appropriative output operator G is activated. From now these aggregates can communicate directly.

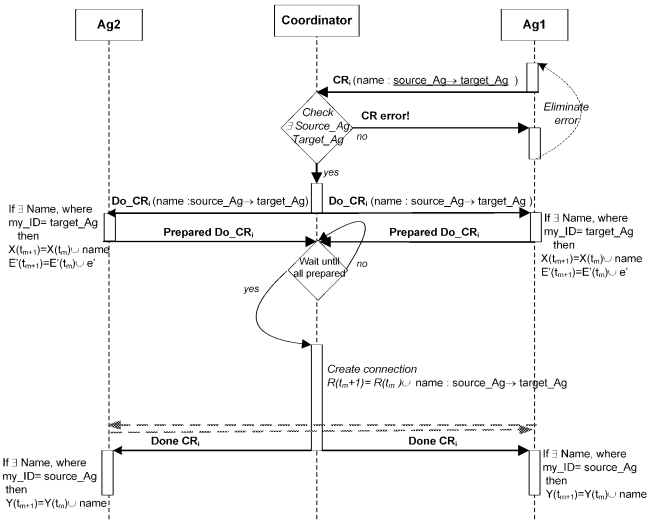


Figure 2: CR protocol

Using the CA protocol, aggregates can create their children aggregates. When an aggregate is created, the coordinator sends the "Start" message to launch it for work. Deleting aggregates, primarily all links are deleted using the DR protocol. After the coordinator gets all the expected "prepared_Do_DR", it sends an extra "Sleep" message for a certain aggregate. The aggregate responds to "Sleep" message with "clearing" its continuous component $z_v(t_m) = \{\infty,\infty,...,\infty\}$ and sending the "Sleep_Done" message back to the coordinator. Just after these actions, the

coordinator deletes the aggregate from its set of aggregates $A(t_{m+1}) = A(t_m) / A_i$.

DPLA FOR AGENT-BASED APPLICATION

An Example

Figure 3 illustrates an example, which is a composition of two similar scenarios "Tileword" (Schattenberg and Uhrmacher 2001) and "Packet world" (Weyns and Holvoet 2002) for testing various features of agents. An example presents a world, which consists of obstacles, different colors packets and holes, which have to be filled with the same color packets.

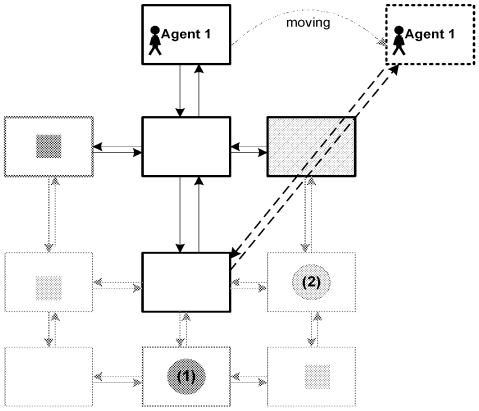


Figure 3: An example of a dynamic environment

The environment is changing in time, which means that elements are changing their locations in the world. An agent has to pick packets and bring them to corresponding holes. It has a map of the world using which it determines intentions for moving to secure one's object. The agent can move on the second position in four directions ($\leftarrow\uparrow\rightarrow\downarrow$) except obstacles. The generation of the action plan and every step of the agent are time consuming. The agent generates his plan by viewing in its map and seeking an object. Whenever it finds that object, the agent generates a plan how to get there in minimum steps. The agent is driven by its goals. The main goal of the agent is to score more points; it is defined by the goal function. The general goal consists of primary sub-goals: find packet, put it to the corresponding hole. These sub-goals form intentions of agents of additional actions, in turn believe about surrounding environment influences its intentions. Goals, believes, intentions are main initial features of the agent. However, in this application the environment is changing and sometimes believes of agents are not trustable. So it is relevant to test believes of agents in a dynamic environment.

DPLA Specification

In DPLA, the described example is taken as one aggregate, where all elements of the world and the agent are its children aggregates $A(t_m) = \langle v_{ij}, Ag \rangle$. All elements of the world v_{ij} have relationships (a set of input X and output Y signals) with v_{kl}, $v_{kl} \left| (k-i)^2 + (l-j)^2 = 1, \quad 1 \le k,l \le n \right.$ neighbor elements. The agent is moving by removing and adding couplings with elements (cp. Figure 3). The agent has couplings only with one element that denotes its position $poz(t_m) = \{a1, a2)$ in the world. The movement of agents in DPLA is described using CR and DR protocols. At first it sends a message $y = DR(y_i : Ag \rightarrow v_{p1,p2}, \quad p1 = a1, p2 = a2$ to its coordinator for deleting couplings with the present element and after $y = CR(y_i : Ag \rightarrow v_{ij}, \quad v_{ij} = move_poz(t)$ for creating couplings with a new element. The map, gained points, believes, intentions, goals are discrete components $v(t_m) = \{map(t_m), P(t_m), bl_a(t_m), bl_b(t_m), I(t_m), ...\}$ of the agent. The generation of plans, intentions of moving are defined as internal events $E''(t_m) = \{e_1'', e_2''\}$. Controlling sequences denote the duration of plan generation $e_1'' \mapsto \{\gamma_i\}$ and one step of the agent $e_2'' \mapsto \{\chi_i\}$.

Each element has a type $d(t_m)$, which consists of three constituents: the color $c(t_m) \in \{red, green, blue\}$, the type $t(t_m) \in \{packet, hole, empty, obs\}$ and capacity $p(t_m) = N$ (if it isn't a hole, then 0). The type of the element is defined in its discrete component $d(t_m) = \langle c(t_m), t(t_m), p(t_m) \rangle$. Depending on input signals received from the agent, this type could be changed. If an element is a packet and if it receives a signal y_{packet} from an agent, it becomes an empty element $c(t_{m+1}) = \varnothing$, $t(t_{m+1}) = empty$, $p(t_{m+1}) = 0$. Each element has one internal event e'' – to change the position. The controlling sequence of the internal event $e'' \mapsto \{\varphi_i\}$ determines how often the element changes its position.

This approach permits using a single mathematical scheme to create models both for simulation and validation. The occurrence graph enabled us to validate specification and to analyze such systems features as accessibility, necessity of continuous coordinates, bound of state coordinates, deadlocks and loops. This validation method is implemented in specification analysis tool PRANAS2, based on the aggregate approach.

Experiments

DPLA is able to specify multi-agent systems, which often consists of several concurrently acting or collaborative agents. To test capabilities of agents for reaching their goals dealing with the dynamic environment, we decided to experiment and to add the second self-learning agent in our application. Such an agent has capabilities to calculate a probability that the object under seeking is in the same position. The results were compared afterwards: the number of incorrect believes and gained points.

For DPLA realization of the presented example we used JAVA programming language. Some results of simulation are presented in figure 4.

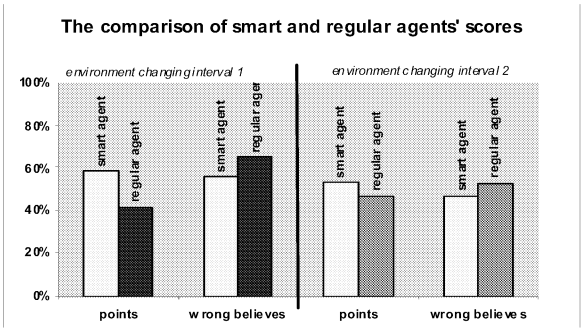


Figure 4: Points and wrong believes of smart A1 and regular A2 agents

In our experiments two agents are concurrently active in the surrounding environment. In our first experiment, where environment is changing in interval 1, a smart agent A1 gets 16,5% more points and makes 9,91% less wrong believes than a regular agent A2. In our second experiment we extended the interval (interval 2) so that the environment is changing double-slower. As we expected the A1 agent becomes less superior: A1 gains 6,55% more points and makes 5,66% less wrong believes than A2 in the environment 10×10, but enlarging the size of the environment to 20×20, the efficiency of A1 increases to 7,74% more points and 13,11% less incorrect believes. From the results we can see that the more often environment is changing, the more useful learning capabilities of a smart agent are, besides enlarging the size of the environment the efficiency of A1 increases as well.

CONCLUSION

In this paper, the dynamic approach of the formal method PLA was presented for formalization of multi-agents systems, were agents and the surrounding environment are described as a system of aggregates. DPLA has enabled us to describe the dynamic structure of MAS. We represented DPLA using a concrete multi-agent application to test capabilities of agents for reaching their goals in dynamic environment. DPLA allows us to describe all features of MAS in a formal, compact and readable description. Furthermore, here is a possibility of formal validation that enables to prove the correctness of the specification. In the future, this approach will be used for formalization of complex systems in real applications.

REFERENCES

d'Inverno, M.; M. Luck, M. Georgeff; D. Kinny; and M. Woolbridge. 2004. "The dMARS architecture: A Specification of the Distributed Multi-Agent Reasoning System". In *Journal of Autonomous Agents and Multi-Agent Systems*, Vol.9, No.1-2 (Jul), 5-53.

Fox, J.; M. Beveridge; and D. Glasspool. 2003. "Understanding intelligent agents: analysis and synthesis". *AI Communications,* Vol. 16, No. 3 (Aug), 139-152.

Luck, M. and M. d'Inverno, 2003. "Unifying Agent Systems". *Annals of Mathematics and Artificial Intelligence,* Vol. 37, No. 1-2 (Jan), 131-167.

Pranevicius, H. 2003. "Formal specification and analysis of Computer Net Protocols: aggregate method". *In the book*, Technologija, Kaunas, Lithuania, 273 pages.

Rohl, M and A.M. Uhrmacher. 2004. "Controlled Experimentation with Agents - Models and Implementations". In *5th International Workshop. " Engineering Societies in the Agents World*". Toulouse, France, (Oct) 20-22.

Schattenberg, B. and A.M. Uhrmacher. 2001."Planning Agents in James". In *Proceedings of the IEEE*, Vol.89, No.2 (Feb), 158-173.

Uhrmacher, A.M. 2001a. "Dynamic Structures in Modeling and Simulation - A Reflective Approach". In *ACM Transactions on Modeling and Simulation*, Vol.11, No.2 (Apr), 206-232.

Uhrmacher, A.M. 2001b. "A System Theoretic Approach to Constructing Test Beds for Multi-Agent Systems". In *Discrete Event Modeling and Simulation Technologies. A Tapestry of Systems and AI-Based Theories and Methodologies: A Tribute to the 6 0th Birthday of Bernard P. Zeigler*. Lecture Notes on Computer Science, New York, Springer, 315-339.

Weyns, D. and T. Holvoet. 2002. "A Coloured Petri Net for a Multi-Agent Application". In *Proceedings of Modeling Objects. Components and Agents, MOCA'02*. Denmark, (Aug) 121-140.

Wooldridge, M. 1999. "Intelligent agents". In *Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, USA, 643 pages.

Xu, D and Y. Deng. 2000. "Modeling mobile agent systems with high level Petri nets". In *Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC'2000),* Vol.5, 8-12 (Oct), 3177-3182.

AUTHORS BIOGRAPHIES

HENRIKAS PRANEVICIUS is professor, dr.habil., the head of the department of Business Informatics at Kaunas University of Technology. Author received his doctor degree at Kaunas Polytechnic Institute in 1970, habilitation at Ryga Electronic and Computer Technical Institute in 1984. His research interests include the formal specification, validation and simulation of complex systems, knowledge-based simulation, and development of numerical models of systems specified by Markov Processes. The results of investigations have been successfully applied in creating the computerized systems for specification, validation and simulation/modeling of computer network protocols, logistics and industrial systems.

DALIUS MAKACKAS is an associated professor in Business Informatics Department at Kaunas University of Technology. His main research interests are formal methods, real time systems and the theory of algorithms.

AGNE PAULAUSKAITE-TARASEVICIENE is a PhD student in Business Informatics Department at Kaunas University of Technology. The title of her thesis is "Modeling and formalization of multi-agent systems using formal method PLA". Student's research interests include formal methods, systems of dynamic structure.

LIVING DESIGN: SIMULATION FOR SELF-DESIGNING AGENTS

Sylvain Lemouzy
Carole Bernon
Marie-Pierre Gleizes
IRIT – Paul Sabatier University
118 route de Narbonne
31062 Toulouse cedex 09
France
E-mail: {lemouzy, bernon, gleizes}@irit.fr

KEYWORDS
Self-organisation, self-adaptation, software engineering.

ABSTRACT

Today's challenge is to design complex systems that operate within evolving environments. Multi-agent systems and self-organisation are an answer to implement such systems. However few tools are provided to designers for helping them to build or improve the behaviour of agents. The aim of the work presented here is to reduce the workload of a MAS designer with a simulation tool that would give agents the ability to self-design their behaviour considering that (i) a behaviour can be expressed as a set of rules and (ii) agents interact in the dynamic environment they are situated in. This paper shows how simulation has improved the behaviour of simulated foraging ants.

INTRODUCTION

Multi-agent systems (MAS) are a recognised paradigm to deal with the complexity and openness of today's computer systems. Furthermore adaptation is a mandatory characteristic of these systems to make them able to stay operational in today's evolving environments. A possible way for a system to achieve this adaptation ability is to self-organise (Di Marzo Serungendo et al. 2005; Heyligen 2003) that is to change its internal organisation without an explicit external control during its execution time.
The work presented in this paper has been done in the context of self-organising systems design, more especially on those that are based on the Adaptive Multi-Agent System (AMAS) theory (Capera et al. 2003) and designed using the ADELFE methodology (Bernon et al. 2005). These two concepts are then detailed hereafter.

Self-organising Systems and AMAS Theory

Let us suppose that agents of a system are endowed with the ability to autonomously and locally modify their interactions for reacting to changes in their environment. These alterations also transform their collective function i.e. the function performed by the multi-agent system they belong to. This latter is a self-organising system and becomes then able to adapt to changes in its environment. This assumption is the foundation of the AMAS theory that considers that "cooperation" is the criterion for self-organisation (Capera et al. 2003). Actually, every agent pursues a local objective and interacts with others while respecting cooperative rules that make it changing its interactions in order to avoid and possibly remove situations called non cooperative situations or NCS) that are judged as non cooperative from its local point of view.

Roughly, applying the AMAS theory consists then on the one hand in trying to anticipate NCS and to avoid them and on the other hand, when NCS cannot be avoided, to process them to come back to a cooperative state. Therefore a designer has to enumerate, according to the current problem to solve, all the cooperation failures that can appear during a MAS functioning and then to define actions cooperative agents have to apply to stay in or to come back to a cooperative state. However, building self-organising systems this way is not so easy and tools are required to help designers enforcing this theory.

ADELFE Methodology

The ADELFE agent-oriented methodology (Bernon et al. 2005) aims at guiding AMAS designers through a development process based on the RUP (Rational Unified Process). This process has been modified to take into account specific features of AMAS, especially those concerning cooperation. In particular, during the final requirement phase, designers are invited to think about the situations that can be "unexpected" or "harmful" for the system because they can lead to NCS at the agent level. These situations can be then expressed in use case diagrams. Modifications in the design phase essentially concern the design of cooperative agents. The most difficult point here is to find local cooperation rules for guiding the behaviour of agents in order to have the best consistent behaviour at the system level. A fast prototyping activity is then provided for testing the behaviour of agents mainly during interaction protocols. However, according to the AMAS theory, designers have to find all NCS an agent may encounter and provide, for every detected NCS, all actions this agent has to do to stay cooperative toward others and itself. A first additional tool was then added during the implementation phase to help designers finding NCS not taken into account yet for an agent. This simulation tool (Bernon et al. 2006) based on SeSAm platform (Klüg et al. 2003) enables designers to observe a prototype of the targeted MAS while running and automatically detects some kinds of NCS. Designers are then able to change accordingly the behaviour of the concerned agents.

These modifications are still done manually and further help would be required to allow an agent to always choose the action which can be considered as the best cooperative one. The work presented here is a first step towards our final objective that is to enhance ADELFE with such a tool.

PROBLEM STATEMENT

The aim of the work presented in this paper is to alleviate the design task by providing designers with a means for automatically designing an agent behaviour considering the following context:

- This behaviour can be expressed with a set of behavioural rules which follow this pattern:
  if premise then consequent
  where premise is a logical predicate made up of elements coming from agent perceptions or characteristics and consequent activates one of the possible actions this agent may perform. All the rules needed to design the decision process are given by a designer that is the agent does not learn new rules during the process and the set of given rules is complete and correctly written.
- The system interacts with a dynamic environment.

Therefore, an agent behaviour consists in executing some rules of a set of executable rules at a given time in the decision process. Modelling this kind of behaviour was inspired by the subsumption architecture (Brooks 1986) in which rules represented by (conditions, actions) tuples are ordered depending on the level of priority of an action relatively to another one. If several rules are executable, an agent executes only the one of highest priority. In our model an agent may execute several rules at the same time. However the priority between the sets of rules, which gives the algorithm executed by an agent, was retained from the subsumption concept.
An agent for which the behaviour has to be enhanced is simulated under SeSAm by reusing the cooperative agent model described in (Bernon et al. 2006) to implement a prototype of the targeted MAS this agent is part of. This agent runs, interacts with its environment and gives its conclusions (drawn from interpreting and reasoning about its perceptions) to its Behaviour Self-Design Module (BSDM). Considering what this agent knows, its BSDM gives it back a list of actions to be executed which corresponds to those of the top-level set of rules of the current hierarchy. This BSDM is implemented as an AMAS, behavioural rules forming it have to collectively adapt to the agent's environment and are then considered as agents. They are called rule-agents for not confusing them with the agent which contains the module, this latter agent is called a simulated agent.

Our aim is to enable the rules inside the BSDM to self-organise in order to find the best hierarchy of rules that is to say the most efficient behaviour for the simulated agent it belongs to. At the beginning of the implementation, the designer supplies the BSDM with all the behavioural rules put in a random order. By forming sets of rules which have to be executed simultaneously and then by prioritizing these different sets according to what its simulated agent perceives, the BSDM provides this agent with the ability to self-design its behaviour.

SELF-DESIGNING AGENT BEHAVIOURS USING SIMULATION

According to the AMAS theory, an AMAS carries out the right collective function if its agents cooperate in the right manner. This section describes the rule-agents making up the self-design module and then how these rules cooperate in order to find their right place in the subsumption-inspired hierarchy presented above and consequently the most efficient behaviour for a simulated agent.

Structure of Behavioural Rules and their Relationships

As said before, the pattern followed by the behavioural rules of a simulated agent can be summarized by: if conditions then actions. (Bernon et al. 2006) considered conditions written with propositional logic formulae, this work has been generalised in order to consider conditions and actions as logical predicates that may be parameterised. A rule is still triggered if all its conditions are satisfied and actions are still described using a set of preconditions (i.e. conditions that trigger them), a set of additional effects (predicates added when these actions are executed) and a set of removal effects (predicates removed when these actions are executed). A rule is then structured in this way:

$$\mathrm{Cond}_1([\mathrm{args}]) \wedge \ldots \wedge \mathrm{Cond}_n(\mathrm{args}]) \rightarrow \mathrm{Action}([\mathrm{args}]) : + \{ \mathrm{Add}_1([\mathrm{args}], \ldots, \mathrm{Add}_m([\mathrm{args}]) \} - \{ \mathrm{Rem}_1([\mathrm{args}]), \ldots, \mathrm{Rem}_p([\mathrm{args}]) \}$$

This structure allows us to identify two types of relations between agents representing these rules:

- A rule-agent R1 interferes with another rule-agent R2 because (1) R1 performs an action which contains effects that are opposite to those of R2 or (2) by executing an action, R1 removes at least one of the conditions of R2. This relation is called inhibition of effects in the first case and is written: InhibitsE(R1, R2); inhibition of activation in the second case which is written: InhibitsA(R1, R2).
- A rule-agent R1 helps another rule-agent R2 because it makes at least one of the conditions of R2 appear. This relation, written Permits(R1, R2), is called permission.

As a simulated agent executes, it gives its conclusions to the BSDM expressed as logical predicates. If conditions of a rule-agent R can be instantiated by these predicates, R may be triggered and this state is written: MayBeTriggered(R). R is actually triggered (and possibly executed by the simulated agent) if when R has to act, depending on its level in the hierarchy, its conditions are still satisfied, this is written: IsTriggered(R).
We also consider that a rule-agent R1 knows whether it is situated above another one R2 in the hierarchy, this is written: Above(R1, R2).

For performance reasons and because the solving process in AMAS is a local one, cooperative agents have only a local view of their environment. Thus, a rule-agent only knows some other rule-agents, called neighbours: those which may

be triggered, those which are actually triggered and the rule-agent which is situated just immediately above it.

Finding the right hierarchy is to find the right organisation between the rule-agents composing the self-design module. By definition this organisation has to emerge from the cooperative interactions between rule-agents that have to consider the above relationships.

Non Cooperative Situations Rule-Agents Encounter

Hints for finding the cooperation rules that govern rule-agents can be discovered considering the assumptions that make the hierarchy being the right one:

- Rules that are triggered have to always be consistent because a simulated agent cannot execute some actions which inhibit each other,
- Rules that have little probability to be executed have to be given priority because it was assumed that every rule is useful,
- Considering that this last point remains satisfied, a maximum number of rules has to be executed making the simulated agent the most efficient possible.

If one of these criteria is not satisfied, at least two rule-agents are in a NCS i.e. the order of the hierarchy is not optimal.

All the kinds of NCS that can theoretically appear in an AMAS (Capera et al. 2003) were studied with respect to these previous assumptions and three potential NCS were identified between rule-agents:

- Theoretically a conflict may appear when two agents have antinomic effects. A conflict occurs then if two rule-agents R1 and R2 are triggered but R1 prevents R2 from being expressed i.e. InhibitsE(R1, R2) $\wedge$ IsTriggered(R1) $\wedge$ IsTriggered(R2).
- Another conflict is possible if R1 is above R2, R2 could be activated but is not triggered and R1 inhibits R2 because it is actually triggered i.e. InhibitsA(R1, R2) $\wedge$ IsTriggered(R1) $\wedge$ MayBeTriggered(R2) $\wedge$ $\neg$ IsTriggered(R2) $\wedge$ Above(R1, R2).
- An NCS of uselessness may occur when an agent believes what it is going to do is not beneficial for others or itself. Such a situation appears if a rule-agent R1 is triggered, another one R2 located above R1 is not and R1 judges that it could have helped R2 to be triggered i.e. IsTriggered(R1) $\wedge$ Above(R2, R1) $\wedge$ $\neg$ IsTriggered(R2) Permits(R1, R2) $\wedge$ $\neg$ InhibitsA(R1, R2) $\wedge$ $\neg$ InhibitsE(R1, R2).

In order to detect possible NCS, a rule-agent learns by observing the relationships it has with its neighbouring rule-agents. To always stay in a cooperative state, this agent has then to remove the NCS it has detected.

Solving NCS by Moving in the Hierarchy

Rule-agents autonomously change their interactions by going up or down the hierarchy in order to avoid or remove NCS. For simplifying their behaviour, these agents can only move from one position to the adjacent one at the same time. Besides, considering that when "R2 is above R1", "R1 moves up just above R2" is equivalent to "R2 moves down just below R1", giving the ability to move in only one direction is enough (e.g. upwards). A rule-agent that succeeds in climbing the hierarchy by only local ascents had then all the legitimate reasons to be there. Therefore a rule-agent is going to move upwards for solving NCS.
If several NCS are encountered by a cooperative agent, this latter is not always able to remove them all and a priority to process them has to be given: a rule-agent tries first to avoid interfering with others and then tries to help them. A rule-agent is then going to try to avoid conflicts before solving situations of uselessness.
Let us suppose now that a rule-agent R1 is above another one R2. If R2 moves above R1, it prevents it from going up. In order to stay the most cooperative possible, a rule-agent can move up only if it has at least as many reasons to go up than the rule-agent just above. Another criterion is then taken into account for allowing rules to move: the priority they have to do this that is inversely proportional to the number of times a rule-agent may be triggered.

To sum up, a rule-agent R1 decides to go up considering the rule R2 just below it as follows:

- If R1 wants to move up because of a conflict and R2 wants also to go up because of a uselessness situation, then R1 has priority over R2; solving a conflict is more vital than removing a uselessness situation.
- If R1 and R are motivated to move up by the same kind of NCS, then the rule with the highest priority does.
- If R1 wants to go up for solving an NCS and R2 has detected no NCS, then R1 is free to move up.

Mechanism of Self-Organisation

The mechanism that makes rule-agents autonomously reorganise their interactions is a loop made up of two steps:

1. In the first one, each agent-rule of the hierarchy which is possibly and/or actually triggered learns the new relationships (detailed in the first paragraph of this section) it has with its neighbours considering the execution of the current hierarchy.
2. In the second step, each agent-rule (from the one on top of the hierarchy to the one at the bottom) which is possibly and/or actually triggered, reasons and decides to move (up) in the hierarchy or to stay where it is depending on the relationships it has previously learned and the NCS it has therefore may be detected.

Because rule-agents have only a local view, they do not have a way for knowing whether the current hierarchy they belong to is the right one i.e. the behaviour they have collectively found is the best one for the simulated agent they are working for. To be stopped, this mechanism needs therefore an external intervention, coming from a designer who observes the simulation and the dynamic structuring of the hierarchy and decides that the obtained hierarchy is steady and efficient enough for forming the behaviour he/she was seeking.

This module was implemented as a plug-in for the SeSAm simulation platform (Klügl et al. 2003). In order to validate

the approach proposed above, experiments were carried out on a MAS simulating a society of foraging ants.

EXPERIMENTS ON BEHAVIOUR OF ANTS

This MAS was implemented using SeSAm and the ant-agents composing it use this module to automatically find their behaviour. The set of rules, in which the order of rules is completely arbitrary, initially given to the simulated ant-agents is the following:

R1. AvailableTime() ∧ FullLoaded() ∧ Nest(X) ∧ On(X)
→ DropFood(X)

R2. AvailableTime() ∧ FullLoaded() ∧ NoCollision() ∧ Nest(X)
→ MoveTowards(X)

R3. AvailableTime() ∧ Foraging() ∧ Food(X)∧ On(X)
→ PickUpFoop(X)

R4. AvailableTime() ∧ Foraging() ∧ NoCollision() ∧ Food(X)
∧ Near(X) → MoveTowards(X)

R5. AvailableTime() ∧ Foraging() ∧ NoCollision()
∧ Pheromone(X) ∧ Near(X) → MoveTowards(X)

R6. AvailableTime() ∧ Foraging() ∧ NoCollision()
→FreeMove()

R7. AvailableTime() ∧ Collision() →DodgeObstacle()

R8. FullLoaded() → DropPheromone()

Where the meaning of the conditions for an ant-agent A are the following:

- AvailableTime() means that A has time to execute an action
- Collision(): A collides with an obstacle (another ant is also considered as an obstacle)
- Foraging(): A forages (searches randomly after food)
- Food(X): X is an item of food
- FullLoaded(): A has picked up some food and cannot pick up more
- Near(X): A has detected X which is near it
- Nest(X): X is a nest of a colony of foraging ants
- NoCollision(): A risks no collision with an obstacle
- On(X): A is located on X
- Pheromone(X): X is some pheromone item

Effects of actions a simulated ant-agent can take are described as follows:

A1. DropFood(X): +{ Foraging() };
-{ AvailableTime(), FullLoaded() }

This means that the effect of dropping food enables the ant to forage again (+Foraging()) and implies the ant has no time to do another action during this step (-AvailableTime()) and it is no more full loaded (-FullLoaded()).

A2. MoveTowards(X): +{ }; -{ AvailableTime() }
Effect of moving towards another location implies the ant has no time to do another action during this step (-AvailableTime()).

A3. PickUpFood(X): +{ FullLoaded() };
-{ AvailableTime(), Foraging(), Food(X), On(X) }

When an ant picks up food this implies that this ant is full loaded (+FullLoaded()), it has no time to do another action during this step (-AvailableTime()), it cannot continue to forage (-Foraging()), the food X exists no more (-Food(X)) and the ant is no more on X (-On(X)).

A4. FreeMove(X): +{ }; -{ AvailableTime() }
Effect of moving anywhere implies an ant has no time to do another action during this step (-AvailableTime()).

A5. DodgeObstacle(X): +{ NoCollision() };
-{ AvailableTime(), Collision() }

This means that the effect of avoiding an obstacle implies there is no more collision (+NoCollision()), the ant has no time to do another action during this step (-AvailableTime()) and the collision is avoided (-Collision()).

A6. DropPheromone(X): +{ }; -{ }
This means that dropping pheromone has no effect.

For example, according to the rule R3 if an ant is foraging and is situated on X where X is recognised as food, then this ant acts (A3) to pick up the food it has found. This makes it full loaded and removes the time allocated to the ant for this step, its foraging state, the food from the environment and the fact that the ant is on a piece of food. Almost all the actions of the given rules consume the time allocated to an ant-agent for the current step, almost all the rules are therefore going to inhibit each other. Some rules are helping each other, for instance, R3 permits R8.

Different experiments in which ant-agents used the self-design module for automatically reorganising their set of rules and improving their behaviour were carried out to evaluate and validate the module implementation. Only two are presented hereafter.

Final Order of Rules

The first experiments done enable us to observe the evolution of the self-organising mechanism and to ensure that the ordered set of rules obtained is stable and functionally adequate with the simulation environment. Because rule-agents have no knowledge of the global system and process, an observer has to decide that the solution has reached a stable and proper state. This observer can estimate that the system is steady enough when the order between rules does not change for around 100 cycles. During a cycle, every rule-agent of the BSDM behaves by detecting NCS and deciding to change its location. The functional adequacy is judged by the observer by analysing the behaviour of one ant-agent in its environment.

The experiment with the same initial unordered list of rules (given at the beginning of the previous section) has been done 50 times. At the end of each series of experiments, it can be seen that rules are ordered in the same way with the exception of the last two rules (R2, R7): {R8}, {R3}, {R4}, {R5}, {R6, R1}, {R2, R7}. This can be explained because rules R2 and R7 have no inhibition link and their order is not significant. We can assume that the solution obtained by the BSDM is quasi-unique. In this list, during a cycle, the set i-1 subsumes the set i (except the first set for which

rules can be executed with all the others) and all the rules in a same set are triggered during the same cycle.

Study of Convergence

The following experiment studies how fast an accurate solution is found. The number of cycles required to have a steady hierarchy is measured for four different situations: when 1, 2, 4 or 8 ants are evolving in the simulated MAS and are simultaneously using the BSDM to learn their own behaviour. Results obtained (see Table 1) show that the learning time (number of needed cycles) is relatively short compared to the number of cycles of a simulation, in the best case, 400 cycles are enough to make ant-agents have a steady behaviour. We can observe that the mean number of cycles diminishes when the number of ants increases, this can be explained by the fact that the more ants there is and the more these ants can be aware of new situations and the faster they can "learn".

Table 1: Number of Cycles before Reaching a Steady Hierarchy

	Simulation with 1 ant	With 2 ants	With 4 ants	With 8 ants
Minimum number	401	410	384	389
Maximum number	2561	1997	1843	1285
Mean number	803	643	584	508

CONCLUSION

Several agent-oriented methodologies (Henderson-Sellers and Giorgini 2005) are offered for guiding MAS designers, however very few focus on adaptive systems and emerging functionalities. However simulation could be a means to alleviate designers' task when building complex multi-agent systems that have to operate in evolving and open environments. Some approaches (Fortino et al. 2006) use simulation for analysing a running system and verifying its behaviour and the ones of agents.

The approach adopted here uses simulation with a similar aim but also for taking part in the design process itself. In this perspective, we proposed to implement a prototype of the targeted MAS under the SeSAm platform in which each (simulated) agent uses a Self-Design Behaviour Module to find its most efficient behaviour. This BSDM is designed, following the AMAS theory, as an adaptive MAS made up of agents that represent the behavioural rules of the simulated agent. An organisation of this AMAS represents a certain behavioural hierarchy and therefore a certain behaviour for the simulated agent that uses the BSDM. The objective of a rule-agent within the BSDM is to find its right place in the behavioural hierarchy i.e. to establish the relationships with other rules that make it stay the most cooperative possible. We studied and enumerated the possible relationships between rule-agents to determine the cooperative attitude of a rule-agent and the actions a rule-agent could do to remove the situations that prevented it from staying cooperative toward others and itself. This has enabled us to describe the self-organisation mechanism used by the BSDM to establish the subsumption-like hierarchy of behavioural rules used by a simulated agent.

The BSDM was implemented as a plug-in for SeSAm and experiments were carried out to validate the proposed approach. Even if more experiments are still needed, preliminary results show that this approach is feasible and can help a designer in three different ways:

- Simulation may make him/her saving time. Behavioural rules can be given in any order enabling him/her to observe their impact on the individual behaviours (agent level) or on the collective one (system level).
- Simulation may assist him/her by proposing a behavioural strategy that is efficient and adapted to the simulated environment.
- Simulation can be used for any application domain. The self-organisation process of rules is not based on their semantics but on their relationships that are automatically detected by the provided plug-in.

REFERENCES

Bernon, C.; Camps, V.; Gleizes, M-P. and Picard, G. (2005) "Engineering Adaptive Multi-agent Systems: the ADELFE Methodology". In (Henderson-Sellers and Giorgini 2005), 172-202.

Bernon, C.; Gleizes, M-P. and Picard, G. (2006) "Enhancing Self-Organising Emergent Systems Design with Simulation". *6th International Workshop Engineering Societies in the Agent World*, October 2006, Dublin, Ireland.

Brooks, R. A. (1986) "A Robust Control System for a Mobile Robot". IEEE *Journal of Robotics and Automation,* 2 (1) :14-23.

Capera, D.; Georgé, J-P.; Gleizes, M-P. and Glize, P. (2003) "The AMAS Theory for Complex Problem Solving Based on Self-organizing Cooperative Agents". Proc. *12th IEEE International Workshops on Enabling Technologies, Infrastructure for Collaborative Enterprises,* 383-388, 9-11 June 2003, Linz, Austria, IEEE Computer Society.

Di Marzo Serugendo, G.; Gleizes, M-P. and Karageorgos, A. (2005) "Self-Organization and Emergence in Multi-Agent Systems". *The Knowledge Engineering Review*, 20(2):165-189, Cambridge University Press, Simon Parsons (Eds).

Fortino, G.; Garro, A.; Russo, W.; Caico, R.; Cossentino, M. and Termine F. (2006) "Simulation-Driven Development of Multi-Agent Systems", Workshop on *Multi-Agent Systems and Simulation*, Palermo.

Henderson-Sellers, B. and Giorgini, P. (Eds.) (2005) *Agent-Oriented Methodologies*, Idea Group Pub.

Heyligen, F. (2003) "The Science of Self-organization and Adaptivity – Knowledge Management, Organizational Intelligence and Learning and Complexity". *The Encyclopedia of Life Support Systems*, Publishers.

Klügl, F.; Herrler, R. and Oechslein, C. (2003) "From Simulated to Real Environments: How to use SeSAm for Software Development". In M. Schillo et al. (Eds.), 1st German Conferences *Multiagent System Technologies*, LNAI 2831, Springer, 13-24.

AGENT-BASED MODELING AND SIMULATION OF COOPERATIVE CONTENT DISTRIBUTION NETWORKS

Giancarlo Fortino, Alfredo Garro, Samuele Mascillaro, Wilma Russo
Dipartimento di Elettronica, Informatica e Sistemistica
Università della Calabria
Via P.Bucci, cubo 41C, 87036 Rende (CS), Italy
E-mail: {g.fortino, garro, mascillaro, w.russo}@unical.it

KEYWORDS
Multi-Agent Systems, Discrete-Event Simulation, Agent Oriented Software Engineering, Content Distribution Networks

ABSTRACT

Content distribution networks (CDNs) have recently become the most adopted solution to efficiently delivery content through the Internet. To improve their performances, new techniques and schemes should be carefully designed and analyzed. This paper proposes the agent-based modeling and simulation of cooperative CDNs which are CDNs with surrogate servers clustered into cooperative groups. The modeling phase relies on statecharts-based agents and event-based interactions which ease the design of the active elements of a cooperative CDN. In particular, three CDN architectures have been modeled: conventional CDN, cooperative CDN with full managed clusters, and cooperative CDN with peer-to-peer-based clusters. Simulation, which is based on the Repast toolkit, allows for the evaluation of the average user perceived latency and cache hit ratio of the modeled architectures. The obtained results show that the proposed cooperative CDN architectures have the potential to be more efficient with respect to conventional CDN architectures.

1. INRODUCTION

Content Distribution Networks (CDNs) have been introduced and extensively used in the Internet as effective solution for improving the performance of content delivery by means of coordinated content replication (Peng 2003). In particular, a CDN manages a geographically distributed set of surrogate servers, located at the network edge, that archive copies of identical content, so that users' requests can be fulfilled by the optimal surrogate servers. In conventional CDN architectures, when a user request is redirected to a surrogate server which is not able to fulfil it by providing the requested content, the surrogate server fetches the requested content from the origin server, which stores all the offered content, and delivers it to the requesting user. As the origin server is usually far way from each surrogate server, this basic scheme to deal with missing content causes a high average user perceived latency. To overcome this issue, several cooperative caching mechanisms and architectures for CDNs have been proposed (Fortino and Russo 2007; Sivasubramanian et al. 2004; Ni and Tsang 2005).

In (Fortino and Russo 2007) two architectures for cooperative CDNs have been proposed in which surrogate servers, grouped into clusters of neighbor surrogate servers, cooperate to provide the requested content. In particular, when a missing content occurs, the surrogate server checks if a surrogate server of the same cluster has the content so as to forward the unfulfilled user request to it; otherwise, if no other surrogate server has the content, the surrogate server contacts the origin server as in the basic schema. Since surrogate servers in the same cluster are much closer with each other than with the origin server, the average user perceived latency could be really improved. Depending on the organization of the surrogate servers in the cluster, two architectures have been defined: Fully Managed Cluster (FMC) and Fully P2P Cluster (FPC).

This paper proposes the agent-based modeling and simulation of these cooperative CDN architectures. Agent-based modeling and simulation (Luck et al. 2004; Rohl and Uhrmacher 2004; Fortino et al. 2006) has been successfully applied in different application domains ranging from telecommunications to biological and economic systems. In particular, while the modeling phase is driven by the Statecharts-based agent-oriented methodology proposed in (Fortino et al. 2004; Fortino et al. 2005), the simulation phase is based on the Repast toolkit (North et al. 2006), one of the most used agent-based simulation framework.

According to our agent-based approach, the key elements of a CDN (user, surrogate server and origin server) are modeled as agents whose behaviour is defined thorugh a variant of the Statecharts formalism (Harel and Gery 1997). The modeled agents are implemented as Java classes by using the Repast toolkit, which was appositely extended to support discrete event simulation. To compare the performance of the conventional CDN architecture with the proposed cooperative CDN architectures, we developed three simulators: ConventionalCDN, FMC and FPC. The results, which were obtained by varying the number of surrogate servers in the cluster and the number of provided objects, show that the proposed cooperative CDN architectures have the potential to be more efficient with respect to conventional CDN architectures in terms of average user perceived latency and cache hit ratio.

The rest of the paper is organized as follows. Section 2 introduces some basic CDN concepts and describes the FMC and FPC architectures. Section 3 is devoted to the modeling phase of the cooperative CDN architecture and, in particular, details the behaviors of the agents representing the surrogate servers. Section 4 presents the simulation phase of the modeled CDN architectures and, in particular, describes the simulation framework based on Repast and the performance evaluation results. Finally conclusions are drawn and directions of future research briefly elucidated.

2. COOPERATIVE CONTENT DISTRIBUTION NETWORKS

A conventional basic CDN architecture, shown in Figure 1, consists of four basic components: client, surrogate server (or surrogate), redirection system and origin server. When a client issues a content request (1), the redirection system selects the most appropriate surrogate (2) and routes the request to it (3, 4). This surrogate serves the request if it has the requested content (5), otherwise it asks the origin server for the content (6) and, once retrieved, sends it to the requesting client (5).

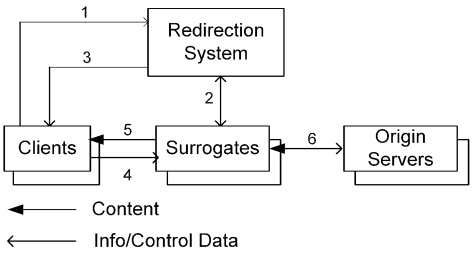


Figure 1: Basic CDN architecture

A missing content (or miss) in the selected surrogate causes a high response time as the origin server is usually located far away from surrogates.

To deal with this issue, cooperative CDN architectures have been envisaged (Fortino and Russo 2007) in which surrogates are grouped into clusters according to their proximity (e.g. neighboring surrogates belong to the same cluster). Surrogates in the same cluster (hereafter called peer surrogates) cooperate to provide a requested content. A surrogate which receives a content request which it cannot serve, forwards such request to a peer surrogate. Only if none of the peer surrogates can provide the requested content, the request is forwarded to the origin server as it happens in a conventional CDN architecture. Cooperation among peer surrogates allows for achieving:

i. a higher hit rate as the available content is not only the content of a single surrogate but the content of all peer surrogates;
ii. a shorter response time as distance between peer surrogates is much shorter than the distance between a surrogate and its origin server.

However, cooperation among peer surrogates demands for the design and implementation of additional mechanisms for cluster management: (i) cluster formation for grouping surrogates in clusters and (ii) cluster maintenance for handling peer surrogates' join/leave.

Two different architectures have been designed: Fully Managed Cluster (FMC) and Fully P2P Cluster (FPC). In both architectures the content stored in a given peer surrogate cannot be duplicated in the other peer surrogates of the same cluster and is, furthermore, always accessible to all peer surrogates of the same cluster. Each peer surrogate has a content location hash table (CLHT) to maintain the content location information for content lookup. Each content has a content identifier (or CId) generated by a collision-free hash function-based algorithm (e.g. SHA-1). In addition, each surrogate has a surrogate identifier (SId) generated with the same algorithm. Moreover, for both the proposed architectures, peer surrogates use an LRU (Least Recently Used) strategy to evict content when storage space is needed.

In the following subsections the FMC and FPC architectures are described.

2.1. The FMC architecture

In this architecture, each peer surrogate manages a CLHT in which stores the content location information of all the peer surrogates. A content request is handled by the selected peer surrogate as follows:

1. if the requested content is in the selected peer surrogate, the content is directly provided to the client;
2. if the requested content is not in the selected surrogate and the CLHT has an entry for that content, the request is forwarded to the peer surrogate which has the requested content and will then serve the client request;
3. if the requested content is not in the selected surrogate and the CLHT has no entry for that content, the request is forwarded to the origin server. In this case, the content will be stored in the selected surrogate before being sent to the client.

The main drawback of this architecture is the overhead for the management of the cluster. In fact, when content is added and/or removed the CLHT of each peer surrogate must be updated. Two update mechanisms of the CLHT were defined: master peer-based and soft-state multicast-based.

2.1.1 Master peer-based

The master peer-based mechanism uses a master peer surrogate (or master) to manage the cluster CLHT. In particular, only the master has the cluster CLHT whereas the other peer surrogates only manage a CLHT of their own content. When a request arrives, a peer surrogate first looks up its CLHT and then, if it does not find the content, forwards the request to the master that, in turn, forwards it either to the peer surrogate (which could also be the master itself) with that content or to the origin server. It is worth noting that every time a peer surrogate chooses to evict a content, it notifies the master that consequently updates the global CLHT. In this way consistency of the cluster is guaranteed by the master even though it could become a bottleneck.

2.1.2 Soft-state multicast-based

The soft-state multicast-based mechanism relies on the soft-state communication paradigm (Raman and McCanne 1997) and BASE (Basically Available, Soft State, Eventual Consistency) semantics (Fox et al. 1997) for which the CLHT could be different in each peer surrogate at any given time. In particular, every update is multicast from the peer surrogate that updated its content to all the others without any coordination mechanism. Each peer surrogate uses such update messages to update its own CLHT. This implies that: (i) according to its CLHT a peer surrogate could forward a request to another peer surrogate that may not have the requested content; (ii) duplicated copies of the same content could be present in a cluster. Although this solution is fully distributed, the consistency of the CLHT is not guaranteed thereby reducing the main advantage of the FMC.

2.2. The FPC architecture

In this architecture, each peer surrogate has an SLT (Surrogate Location Table) which contains the location information of all the peer surrogates and their respective contents. In particular, for each peer surrogate an SLT has an entry formalized by the pair <SId, CZ>, where CZ (Content Zone) is the space of the identifiers of the content potentially stored in the peer surrogate identified by SId. Such an organization implies that a given content can only be stored in the peer surrogate responsible for such content, i.e. its CId is in the CZ of the peer surrogate. According to the FPC model, a content request issued by a client is served by its designated surrogate as follows:

1. if the CId of the requested content belongs to its CZ, the content is looked up in the CLHT. If the content is present, it is sent to the client; otherwise, the content is retrieved from the origin server, sent to the client and finally stored.
2. if the CId of the requested content does not belong to its CZ, the request is forwarded to the peer surrogate responsible for CId which operates as the selected surrogate (see point 1).

The FPC architecture overcomes the main drawback of the FMC architecture as a peer surrogate does not need to maintain content information belonging to the other peer surrogates. However this model has the following drawbacks: since a given content can only be served by a given peer surrogate, a highly requested content can only be provided by a given peer surrogate leading to a service hotspot.

Furthermore, since the mapping between content and surrogate is fixed, when content is added new CIds must be generated and mapped to given zones. This can lead to the problem of zone saturation which demands for a rearrangement of the zones in the cluster. However, if the rate of content update is slow this does not introduce significant overhead.

3. STATECHARTS-BASED MODELING

The cooperative CDN architectures described in section 2 are modeled through an agent-based approach which allowed for the identification of the following agents (see Figure 2):

- *Client Agent* (CA), which represents the client component whose task is to issue content requests according to its needs;
- *Content Origin Agent* (COA), which represents the origin server archiving all content provided by a CDN;
- *Surrogate Agent* (SA), which represents the surrogate server providing the content to clients;
- *Request Redirection Agent* (RRA), which represents the redirection system routing a client request to the selected surrogate.

While the CA, RRA and COA are always the same, different kinds of SA are defined for the different CDN architectures:

- *Conventional SA* (CSA), which represents the surrogate server of a conventional CDN architectures;
- *Full Managed Cluster SA* (FMCSA), which represents the basic peer surrogate in the FMC architecture (see Section 2.2). The FMCSA is further specialized in:
  - FMC Master SA (FMSA), which represents the peer surrogate playing the role of master in the master peer-based FMC architecture (see Section 2.1.1);
  - FMC Slave SA (FSSA), which represents the peer surrogate playing the role of simple peer (or slave) in the master peer-based FMC architecture (see Section 2.1.1);
  - FMC Multicast SA (FMUSA), which represents the peer surrogate in the soft-state multicast-based FMC architecture (see Section 2.1.2);
- *Full P2P Cluster SA* (FPCSA), which represents the peer surrogate in the FPC architecture (see Section 2.2).

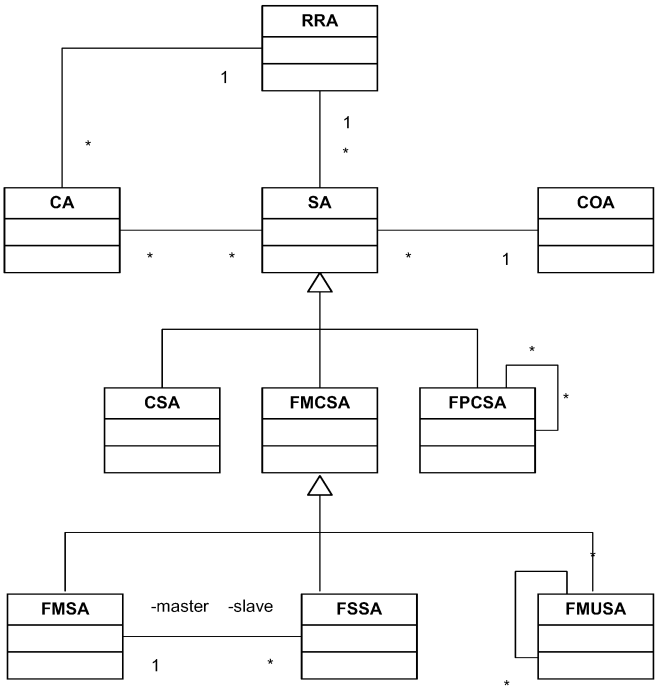


Figure 2: The Agent class diagram

The associations among the agents of Figure 2 are realized through asynchronous events which are reported in Table 1.

The different kinds of SA are all multithreaded agents composed of two types of concurrent behaviours: the server behavior and the front-end behavior (see Figure 3). Upon receiving a ContentRequest event the ServerBehavior activates a Front-EndBehavior which will be in charge to handle this ContentRequest. The generic ServerBehavior (GSB) is defined in Figure 4 through the Distilled Statecharts formalism (Fortino et al. 2004; Fortino et al. 2007). In particular, upon receiving the ContentRequest event, the ServerBehavior executes the action chain ac1: a front-end behavior is spawn (line 2); the target of the ContentRequest event (cR) is changed to the new target, i.e. the front-end identifier (fE) (line 3); the event is routed to the front-end through the *generate* primitive which raises the event (line 4); the RequestDelegated self-event is generated

to bring the ServerBehavior back into the Wait4Request state (line 5).
In the following subsections, we present the formalization of the front-end behavior of the different kinds of SA (CSA, FMSA, FSSA, FMUSA and FPCSA) through the Distilled Statecharts formalism.

Table 1: Asynchronous events exchanged among agents

EVENT	SOURCE	TARGET	DESCRIPTION
ContentRequest	CA	SA	A content request issued by a client to the selected SA
	SA	SA	A content request forwarded by the selected SA to the SA which has the requested content
	SA	COA	A content request forwarded by the selected SA to the origin server
Content	SA	CA	The content requested by the CA
	COA	SA	The missing content requested by the selected SA
	FMSA	FSSA	The missing content requested by the selected slave SA to its master
ContentServed	FMSA	FSSA	The notification of "content served " sent from the master SA to the selected slave SA when another slave SA fulfils the content request
UpdateCLHT	FSSA	FMSA	The event which triggers the transaction for updating the CLHT of the master SA
CLHTUpdated	FMSA	FSSA	The notification that the master SA sent to the slave SA at the end of the update transaction
CLHTUpdated RemoteBroadcast	FMUCSA	FMUCSA	The notification of CLHT update which is multicast to all peer SAs in a cluster for updating their CHLT

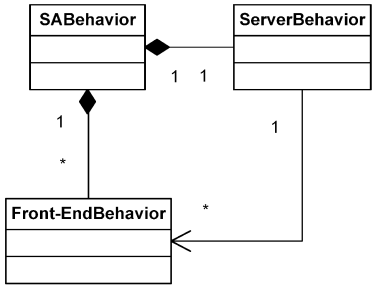
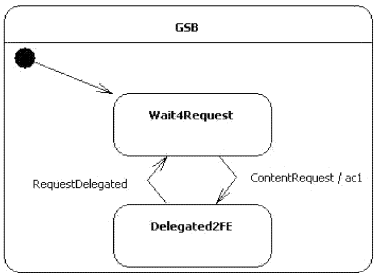


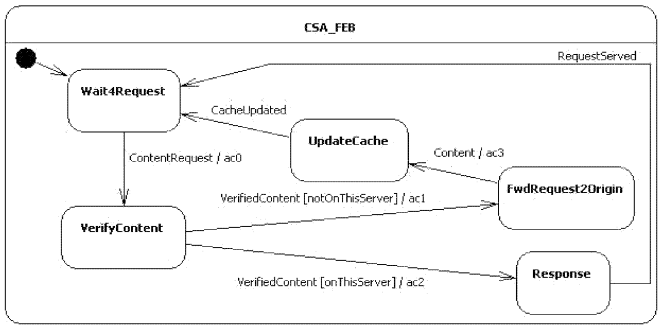
Figure 3: The structure of the generic SA behavior




```
ac1:
1: ContentRequest cR= ((ContentRequest)e);
2: AgentId fE= getFrontEnd();
3: cR.setTarget(fE);
4: generate(cR);
5: generate(new RequestDelegated(self(), self()));
```
Figure 4: The generic ServerBehavior

3.1 The CSA behavior

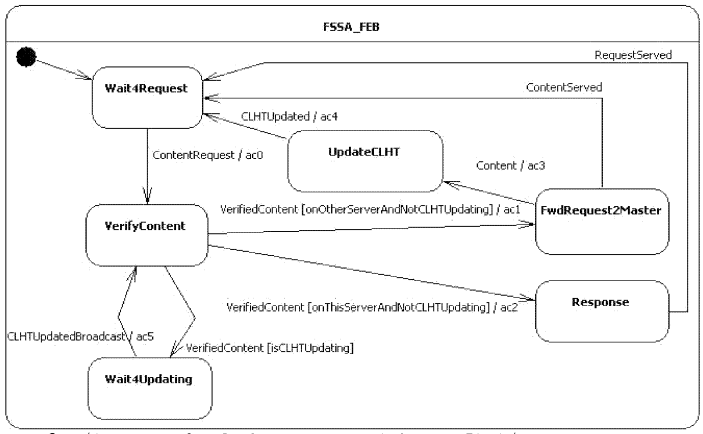
The front-end behavior of the CSA (CSA_FEB) is shown in Figure 5. Upon receiving the ContentRequest event, ac0 is executed which checks if the requested content is available on the server (line 4). If the content is available (i.e. the onThisServer guard holds) the content is served to its requester client identified by idRequester (see ac2); otherwise, the request is forwarded to the OA (see ac1). In the latter case, the OA responds to the front-end with the Content event containing the requested content which is then cached into the server (see action ac3).




```
ac0:
1: ContentRequest cR= ((ContentRequest) e);
2: idRequester=cR.getSource();
3: idContent=cR.getIdContent();
4: contentOnThisServer= isContentOnServer(idContent);
5: generate(new VerifiedContent(self(), self()));
ac1:ContentRequest cR = new ContentRequest( self(),
      originServer, idContent, idRequester);
ac2:
1: Object content = getContent(idContent);
2: generate(new Content(self(), idRequester, content));
3: generate(new RequestServed(self(), self()));
ac3:
1: Object content=((Content)e).getContent();
2: cacheContent( content , idContent  );
3: generate(new CacheUpdated(self(),self());
```
Figure 5: The front-end behavior of the CSA

3.2 The FSSA and FMSA behaviors

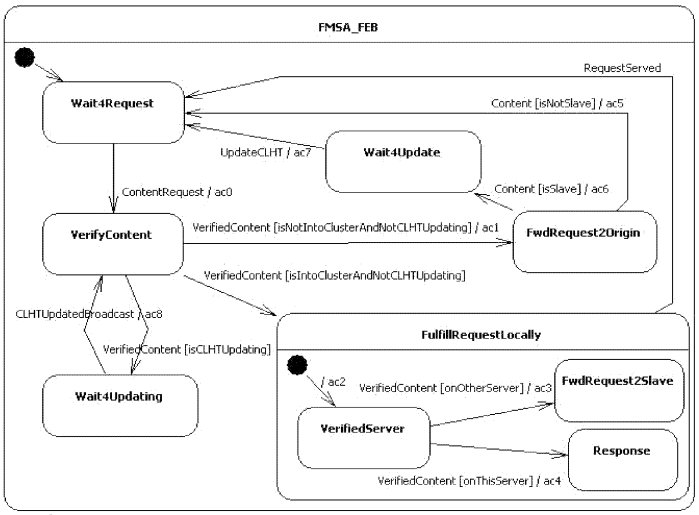
The front-end behavior of the FSSA (FSSA_FEB), shown in Figure 6, is an extension of the CSA_FEB. In particular, in the VerifyContent state, three mutually exclusive transitions labeled by the VerifiedContent event can fire depending on the associated guards: (i) if the content is on the server and the CLHT is not being updated, the content is directly sent to the client; (ii) if the content is on another server and the CLHT is not being updated, the content request is forwarded to the master peer agent; (iii) if the CLHT is being updated the FSSA_FEB goes into the Wait4Updating state.




```
ac0: /* see ac0 of the CSA_FEB (Figure 5) */
ac1:ContentRequest cR= new ContentRequest(self(), master,
                       idContent, idRequester);
ac2: /* see ac2 of the CSA_FEB */
ac3:
1:Object content=((Content)e).getContent();
2:generate(new Content(self(), idRequester, content));
3:String idContentEvicted=cacheContent(content,idContent);
4:generate(new UpdateCLHT(self(), masterServer,
                       idContentEvicted );
ac4: generate(new CLHTUpdatedBroadcast(self(),selfAll()));
ac5: generate(new VerifiedContent(self(), self()));
```
Figure 6: The front-end behavior of the FSSA

In the case (ii), the content can be either served by another surrogate (see transition labeled by the ContentServed event)

or the content is received from the master peer agent and ac3 is executed: the content is sent to its requester client, it is then cached and another content evicted, and finally the UpdateCLHT event is sent to the master peer agent to notify the CLHT update. When the master agent receives the UpdateCLHT event finalizes the CLHT update transaction by sending back to the slave agent the CLHTUpdated event which causes the execution of ac4 in which the slave send the CLHTUpdatedBroadcast event to all the local slave front-ends so waking them up.



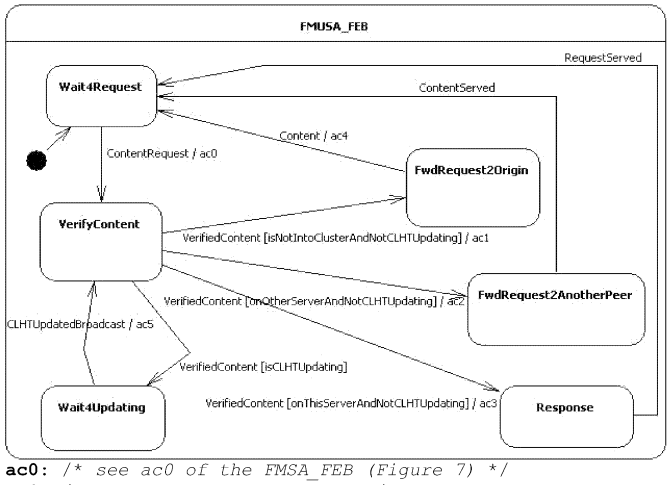

```
ac0:
1: ContentRequest cR= ((ContentRequest) e);
2: slave= isSlave( cR.getSource());
3: if (slave) idSlave=cR.getSource();
4: idRequester=cR.getIdRequester();
5: idContent=cR.getIdContent();
6: generate(new VerifiedContent(self(), self()));
ac1:
1: updatingCLHT=true;
2: generate (new ContentRequest(self(), originServer,
                                idContent, idRequester));
ac2:
1: contentOnThisServer= isContentOnServer( idContent  );
2: generate(new VerifiedContent(self(), self()));
ac3:
1: AgentId slaveServer= getServer( idContent  );
2: generate(new ContentRequest(self(), slaveServer,
                          idContent, idRequester));
3: if (slave) generate(new ContentServed(self(),idSlave));
4: generate(new RequestServed(self(), self()));
ac4:
1: Object content= getContent( idContent  );
2: generate(new Content(self(), idRequester, content));
3: if (slave) generate(new ContentServed(self(),idSlave));
4: generate(new RequestServed(self(), self()));
ac5:
1: Object content= ((Content) e).getContent();
2: generate(new Content(self(), idRequester, content));
3:String idContentEvicted=cacheContent(content,idContent);
4: updateCLHT( idContent , idContentEvicted , self());
5: updatingCLHT=false;
6: generate(new CLHTUpdatedBroadcast(self(), selfAll()));
ac6:
1: Object content= ((Content) e).getContent();
2: generate(new Content(self(), idSlave, content));
ac7:
1: UpdateCLHT update=((UpdateCLHT) e);
2: String idContentEvicted=update.getIdContentEvicted();
3: updateCLHT( idContent, idContentEvicted , idSlave);
4: updatingCLHT=false;
5: generate(new CLHTUpdated(self(), idSlave));
6: generate(new CLHTUpdateBroadcast(self(), selfAll()));
ac8: generate(new VerifiedContent(self(), self()));
```

Figure 7: The front-end behavior of the FMSA

The front-end behavior of the FMSA (FMSA_FEB), shown in Figure 5, is an extension of the FSSA_FEB. The content request (see ac0) can be either a client request (slave is false) or a slave peer request (slave is true). If the content is on the master, it is served (see ac4); otherwise, if the content is in the cluster but not on the master, the content request is forwarded to the slave peer having the content (see ac5). If the content is not in the cluster, the master peer agent requests it to the origin (see ac1). When the origin sends back the content in the Content event, the content is either directly sent to the client if the content requester is not a slave (see ac5) or sent to the slave (see ac6). In the latter case, the CLHT update transaction must be fulfilled (see ac7). In both cases, the CLHTUpdatedBroadcast event is sent to all the local master front-ends so waking them up (see ac5:line 6 and ac7:line 6).

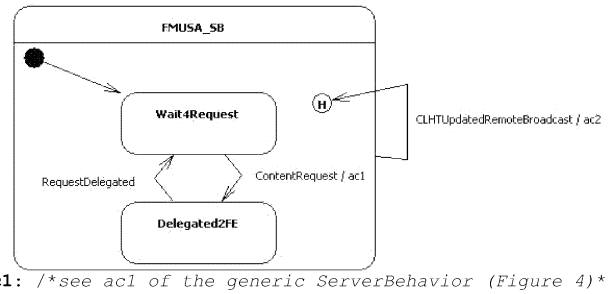
3.3 The FMUSA behavior

The front-end behavior of the FMUSA (FMUSA_FEB), shown in Figure 8, is a variant of the FMSA_FEB. Due to the absence of the master peer agent, a content update is communicated to all the peer agents (see ac4:line 8) through the CLHTUpdatedRemoteBroadcast event. In particular, such event is received by the server behavior of the FMUSA as shown in Figure 9.




```
ac0: /* see ac0 of the FMSA_FEB (Figure 7) */
ac1: /* see ac1 of the FMSA_FEB */
ac2:
1: AgentId idDelegatedPeer= getServer( idContent  );
2: generate (new ContentRequest(self(),idDelegatedPeer,
                                idContent, idRequester));
3: generate (new ContentServed(self(), self()));
ac3: /* see ac2 of the CSA_FEB */
1:
ac4:
1: Object content= ((Content) e).getContent();
2: generate(new Content(self(), idRequester, content));
3:String idContentEvicted=cacheContent(content,idContent);
4: updateCLHT( idContent , idContentEvicted , self());
5: updatingCLHT=false;
6: generate(new CLHTUpdatedBroadcast(self(), selfAll()));
7: String [] md = {idContent, idContentEvicted }
8: for(int i=0; i<peerList.size(); i++){
    generate(new CLHTUpdatedRemoteBroadcast(self(),
                                peerList.get(i),md));}
ac5: generate(new VerifiedContent(self(), self()));
```

Figure 8: The front-end behavior of the FMUSA

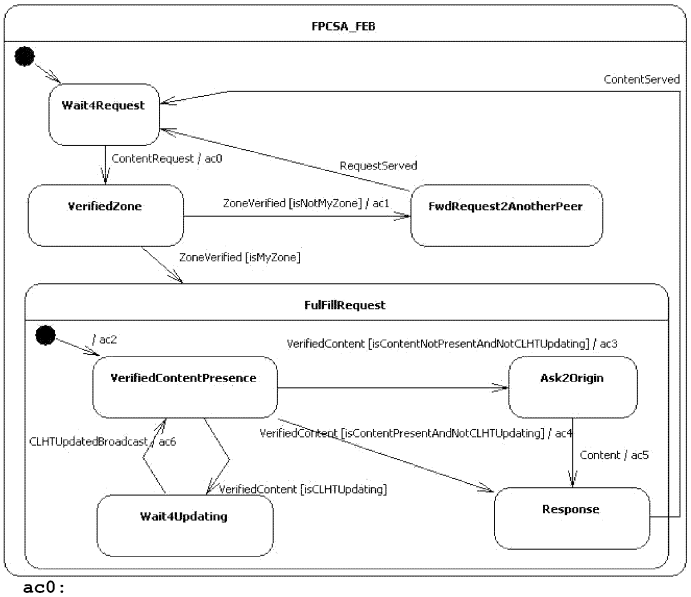



```
ac1: /*see ac1 of the generic ServerBehavior (Figure 4)*/
ac2:
1: String [] mod=((CLHTUpdatedRemoteBroadcast)e).getMod();
2: updateCLHT( mod );
```

Figure 9: The server behavior of the FMUSA

3.4 The FPCSA behavior

The front-end behavior of the FPCSA (FPCSA_FEB) is shown in Figure 10. The content zone (see Section 2.2) is first checked: if the content belongs to its content zone, the content request is fulfilled (see FulFillRequest composite state); otherwise, the content request is forwarded to the peer having competence of the content zone (see ac1). In the former case, the presence of the content is checked: if the content is present, it is sent to the client (see ac4); if the content is not present it is requested to the origin (see ac3) and, once obtained, sent to the client (see ac5); if the CLHT is being updated the front-end waits for the end of the update transaction.




```
ac0:
1: idContent=((ContentRequest) e).getIdContent();
2: idRequester=((ContentRequest) e).getIdRequester();
3: idBelongingZone=verifyZone(  idContent  );
4: generate(new ZoneVerified(self(), self()));
ac1:
1: AgentId peerServer= getServer( idContent  );
2: generate (new ContentRequest(self(), peerServer,
                                idContent, idRequester));
3: generate(new RequestServed(self(), self()));
ac2: contentPresence= lookUp4Content( idContent  );
ac3:
1: generate( ContentRequest(self(), originServer,
                            idContent,idRequester));
2: updatingCLHT =true;
ac4: /* see ac2 of the CSA_FEB */
ac5:
1: Object content= ((Content) e).getContent();
2: generate(new Content(self(), idRequester, content));
3:String idContentEvicted=cacheContent(content,idContent);
4: updateCLHT( idContent , idContentEvicted , self());
5: updatingCLHT =false;
6: generate(new CLHTUpdatedBroadcast(self(), selfAll()));
7: generate(new ContentServed(self(), self()));
ac6: generate(new VerifiedContent(self(), self()));
```

Figure 10: The front-end behavior of the FPCSA

4. SIMULATION

The simulation phase aims at evaluating the proposed cooperative CDN architectures and the conventional CDN architecture. To this purpose, three simulator programs which incorporate the agents described in section 3 are implemented by using the Java-based RePast simulation framework, a well-know simulation framework for agent-based simulations (Repast 2007):

- ConventionalCDN, which simulates a conventional CDN architecture (see Section 2);
- FMC, which simulates an FMC CDN architecture (see Section 2.1);
- FPC, which simulates a master peer based FPC CDN architecture (see Section 2.2).

The developed simulators are based on the following assumptions:

- only one cluster is considered;
- the redirection system redirects a content request to a surrogate of the cluster which is randomly selected among the available ones of the cluster;
- all the caches of the surrogates are equally dimensioned and completely filled at initialization time;
- the number of origin servers is one and this origin server is assumed to be located far away from clients and surrogates;
- average latency times between all clients and origin server are the same;
- average latency times between all surrogates and origin server are the same;
- average latency times between all surrogates of the same cluster are the same;
- average latency times between all clients and surrogates of the same cluster are the same;
- all the objects offered by the CDN architectures have same dimension;
- the processing and latency times are distributed according to an exponential pdf.

In the following subsections the developed Repast-based simulation framework is briefly introduced (Section 4.1), the performance indices and the simulation parameters are defined (Section 4.2), and, finally, some simulation results are presented (Section 4.3).

4.1 Simulation framework

To easily support discrete event simulation with Repast (Repast 2007) the abstract class TAsyncAgent was defined which extends the basic class AsynchAgent (see Figure 11). In particular, the following methods are provided:

- The *setTimer* method (available in two versions) allows for the scheduling of an event to occur after the expiration of the *timeout*. The event corresponds to the invocation of the *method* (without parameters or with parameter *param*) of the agent on which *setTimer* was invoked.
- The *getCurrentTime* method returns the current time of the simulation.
- The *getReport* abstract method should be implemented to report performance measurements of the agent during and after simulation.

All the agents described in section 3 are therefore implemented as (direct or indirect) extension of the TAsyncAgent class.
The three simulators introduced in section 4 are implemented as extensions of the basic ModelImpl class offered by Repast and incorportate their respective agents as well as a Repast-based Space object for the representation of

the network in which the agents are located. Such simulators not only provide the values of the performance indices at the end of the simulation execution but also graphically show them during simulation. Figure 12 portrays the graphical windows of the considered performance indices (see Section 4.2) during simulation.

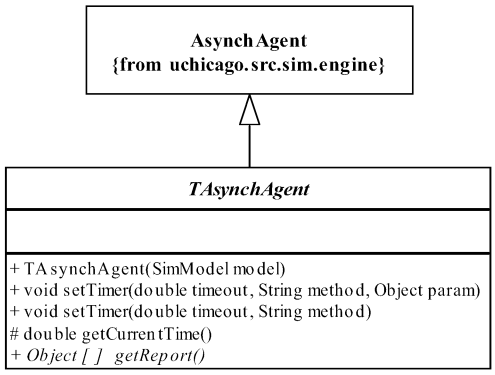
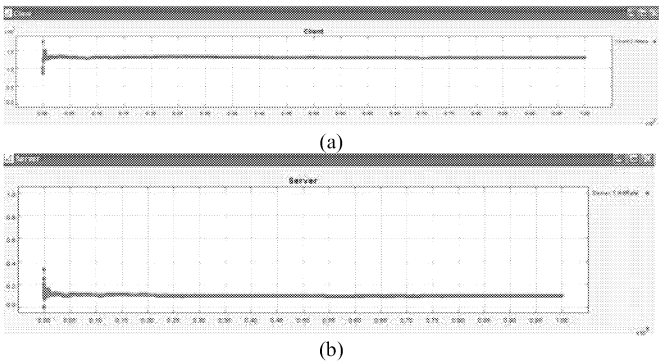


Figure 11: The TAsyncAgent class.



(a)

(b)

Figure 12: Repast graphical windows of the performance indices: (a) average user perceived latency of a client; (b) cache hit ratio of a surrogate.

4.2 Performance indices and simulation parameters

The performance indices on which simulation is focused and that characterize the performances of CDN architectures (Pathan and Buyya 2007) are:

(i) the *average user perceived latency* (AUPL), which is defined as the average time ranging from the transmission of the content request by a client and the reception of the content;

(ii) the *cache hit ratio* (CHR), which is defined as the percentage of content requests successfully served by the CDN without fetching content from the origin server.

The simulation parameters and their considered values are reported in Table 2.

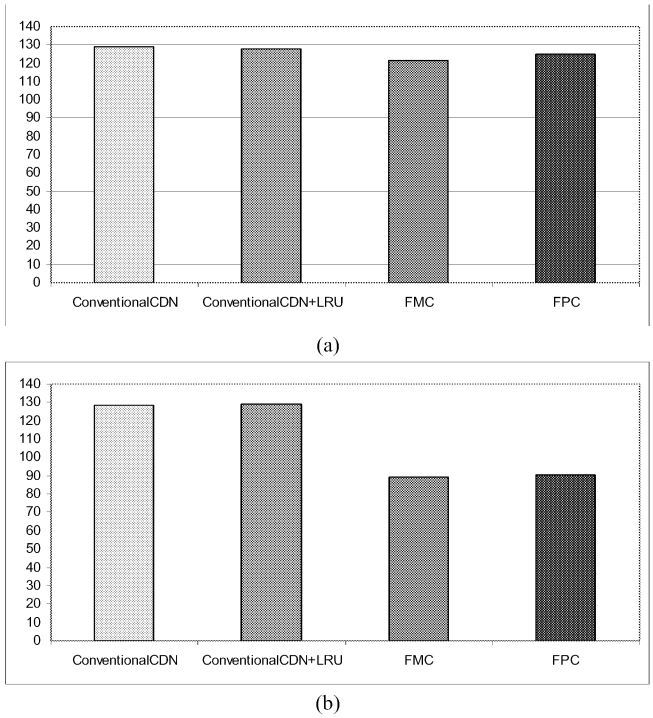
4.3 Simulation results

The obtained results show that the FMC and FPC models are more convenient than conventional CDN models for both the considered performance indices (AUPL and CHR). Of course, the higher is the number of surrogates per cluster, the more efficient is the cooperative approach.

In particular, Figures 13 and 14 report simulation results computed with *Origin Server Objects Number* = 500 and *Cluster Surrogate Number* = 2 (Figures 13a and 14a) and 6 (Figures 13b and 14b). The results regarding the ConventionalCDN architecture are shown both without object eviction from surrogate caches (ConventionalCDN) and with object eviction based on an LRU policy (ConventionalCDN+LRU). The same eviction policy was also employed for the object eviction in the FMC and FPC architectures.

Table 2. Simulation parameters

PARAMETER	VALUE	DESCRIPTION
Origin Server Objects Number	100..1000	The number of objects which are contained in the origin server
Surrogate Object Load	10%	The percentage of objects that are stored in a surrogate with respect to the objects stored in the origin server
Cluster Surrogate Number	2, 4, 6, 10	The number of surrogates in the cluster
Client Average Processing Time	10ms	The average processing time of the client to handle surrogate responses
Surrogate Average Processing Time	10ms	The average processing time of the surrogate to handle content requests
Origin Average Processing Time	10ms	The average processing time of the origin server to handle content requests
Client/Sur Average Latency Time	10ms	The average latency time between clients and surrogates of the same cluster
Sur/Sur Average Latency Time	10ms	The average latency time among surrogates of the same cluster
Sur/Origin Average Latency Time	90ms	The average latency time between surrogates and origin server
Client/Origin Average Latency Time	100ms	The average latency time between clients and origin server



(a)

(b)

Figure 13: AUPL (ms): (a) 2 surrogates per cluster; (b) 6 surrogates per cluster.

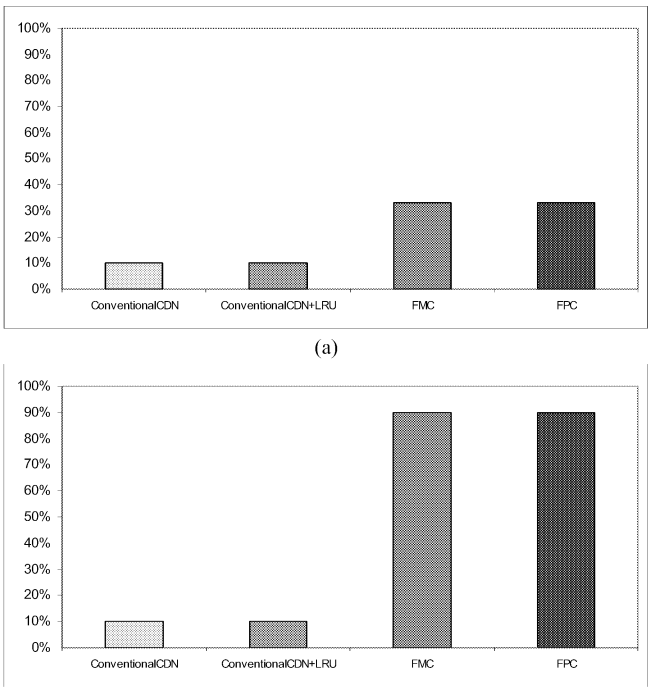


Figure 14: CHR (%): (a) 2 surrogates per cluster; (b) 6 surrogates per cluster.

4. CONCLUSION

Agent-based modeling and simulation is a powerful tool for easily modeling and simulating complex systems. In this paper we have presented the modeling and simulation of cooperative CDNs which were purposely defined to improve performances of conventional CDNs. Modeling relies on Statecharts-based agents whereas simulation on the Repast toolkit. This experience has highlighted two main facts: (i) the modeling and simulation tools employed have really facilitated the design, implementation and evaluation of the proposed CDN architectures; (ii) the cooperative FPC and FMC architectures have the potential to improve performances with respect to conventional CDN architectures. Research efforts are currently geared at evaluating the proposed architectures in more complex simulation scenarios.

REFERENCES

Fortino, G.; A. Garro; and W. Russo. 2005. "An Integrated Approach for the Development and Validation of Multi Agent Systems". In *Computer Systems Science & Engineering*, 20(4), pp. 94-107, CRL Publishing Ltd., Leicester (UK), Jul.

Fortino, G.; A. Garro; S. Mascillaro; and W. Russo. 2007. "ELDATool: A Statecharts-based Tool for Prototyping Multi-Agent Systems". In *Proc. of the Workshop on Objects and Agents* (WOA'07), Genova (Italy), Sept. 24-25.

Fortino, G.; A. Garro; W. Russo; R. Caico; M. Cossentino; F. Termine. 2006. "Simulation-Driven Development of Multi-Agent Systems". In *Proc. of the Workshop on Multi-Agent Systems and Simulation (MAS&S'06), Industrial Simulation Conference (ISC)*, Palermo (Italy), Jun. 5-7.

Fortino, G.; W. Russo; and E. Zimeo. 2004. "A Statecharts-based Software Development Process for Mobile Agents". In *Information and Software Technology*, 46(13), pp.907-921, Elsevier, Amsterdam, The Netherlands.

Fortino, G. and W. Russo. 2007. "Using P2P, GRID and Agent Technologies for the Development of Content Distribution Networks", In *Future Generation Computer Systems*. doi:10.1016/j.future.2007.06.007.

Fox A.; S.D. Gribble; Y. Chawathe; E.A. Brewer; and P. Gauthier. 1997. "Cluster-Based Scalable Network Services," Proceedings of the 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, Oct. 5-8.

Harel, D. and E. Gery. 1997. "Executable Object Modelling with Statecharts". *IEEE Computer*, 30(7), pp. 31-42.

Luck, M.; P. McBurney; and C. Preist. 2004. "A Manifesto for Agent Technology: Towards Next Generation Computing". *Autonomous Agents and Multi-Agent Systems*, 9(3), pp. 203-252.

Ni J. and D.H.K. Tsang. 2005. "Large-Scale Cooperative Caching and Application-Level Multicast in Multimedia Content Delivery Networks," IEEE Communications, 43(5), pp.98-105 , May.

North, M. J.; N.T. Collier; and J.R. Vos. 2006. "Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit". In *ACM Transactions on Modeling and Computer Simulation*, Vol. 16, No. 1, January, Pages 1–25.

Pathan A-M. K. and R. Buyya. 2007. "A Taxonomy and Survey of Content Delivery Networks," Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, Feb. 12. http://www.gridbus.org/reports/CDN-Taxonomy.pdf

Peng, G. 2003. "CDN: Content Distribution Network" *Technical Report TR-125*, Experimental Computer Systems Lab, Stony Brook University.

Raman S. and S. McCanne. 1999. "A model, analysis, and protocol framework for soft state-based communication," ACM SIGCOMM Computer Communication Review, 29(4), pp. 15-25.

Repast (Recursive Porus Agent Simulation Toolkit), software and documentation at the world wide web: http://repast.sourceforge.net/, 2007.

Rohl, M. and A.M. Uhrmacher. 2004. "Controlled Experimentation with Agents - Models and Implementations" In *Proc. of 5th Int'l Workshop Engineering Societies in the Agents World*, Toulouse (France), Oct 20-22.

Sivasubramanian, S.; B. van Halderen; and G. Pierre. 2004. "Globule: a User-Centric Content Delivery Network.," In *Proc of the 4 th International System Administration and Network Engineering Conference*, Sept.

MULTI-AGENT SYSTEMS AND MODELS

A MULTIAGENT VIEW OF GRID COMPUTING FOR BIOLOGICAL SIMULATIONS IN VIRTUAL REALITY

Mikaël Bourhis
European Center for Virtual Reality
25 rue Claude Chappe,
BP 38, F-29280 Plouzané, France
E-mail: bourhis@enib.fr

Vincent Rodin
University of West Brittany (Brest)
20 avenue Victor Le Gorgeu,
29238 Brest Cedex 3 - France
E-mail: vincent.rodin@univ-brest.fr

KEYWORDS

multiagent system; grid computing; biological simulations

ABSTRACT

The grid computing allows to carry out distributed applications. It is a means to make calculations and simulations more efficient than on one computer only. At the CERV (European Center for Virtual Reality), we want to make our simulations of virtual reality applied to biology on a grid of computers. As there is always interactions in these virtual worlds, the distribution is not easy. We want to produce the most autonomous possible grid and a grid used for all our simulations. For that, we choose the agent paradigm that offer to us the concepts and the capabilities of autonomy, self-organization, collaborations, ... This article presents our model called DIVA and an example of application.

INTRODUCTION

Computer simulations are frequently used to calculate and test numerical models. Comparatively to measures and tests achieved on real system, computer simulations offer time and cost gains. Sometimes, it is not possible to obtain these measurements directly on the real system. These simulations can need lots of computing resources and can use cluster or grid computing.
Our research team E.B.V. (Virtual Ecosystemic and Biology) of the L.I.Sy.C. (Computer Science Laboratory of Complex Systems) builds and simulates biological models. In addition to biological calculations, our simulations have realistic visual effects of the simulated biological system. We realize Virtual Reality simulations named *In Virtuo* experiments.

Today, the size of our simulated models are limited. In fact, we use only one computer. The aim of our work is to build a conceptual solution and a concrete solution to the following problem: overtake the resources limitation of the computer system accommodating our simulations. There are several dozens of computers in our laboratory. We want to group them into a grid. Thus, our simulation system will use the whole of resources of this grid. Moreover, this model of grid will be multiagent oriented.

But before the presentation of our first results, we are going to give first of all an overview of the grid computing in the first section. This state of the art, indispensable for the conception of our software, will end with the presentation of the Peer to Peer architecture. In the next section, we will talk about agent paradigm because we want to treat grid computing as a MultiAgent System. We will give details on the properties of the agents and MAS. Next, we will emphasize the transition of the grid nodes status from Peer to Agent. After, we will present our software for the distribution of our biological simulations, and for the end, we will show our first results.

STATE OF THE ART: GRID COMPUTING

We present the different types of grid computing, the main functions of these grids and the two types of encountered architectures.

Taxonomy

The grid computing respecting some demands [Foster *et al.*, 2001] can be classified in two groups: the *global computing* and the *metacomputing*. The *global computing* consists in recovering free CPU cycles of Internet connected personal computers. So, these personal computers connect to a server. The server send them a task to run. The *global computing* uses a trivial parallelism : no communication is necessary between the tasks. For instance, some developed softwares are *SETI@home*, *Decrypton*.
As for *metacomputing*, it uses specific computers, often located in research centers. The set of these computers is very heterogeneous : personal computers and supercomputers. The *metacomputing* is based on the following principle: a client having a problem to solve passes on it to available and competent servers. Several middlewares are used today: DIET , Globus for example. A big size application is divided into a set of elementary tasks. These tasks are linked by dependences (of anteriority or posteriority). These dependences allow to schedule the

tasks into a running stream. Only the independent tasks will be actually executed with parallelism.

Functions

Whatever the considered grid, some functions are always found:

Performance prediction: it is the calculation of the execution time for a problem resolution. The aim is to minimize the resolution global time. This time depends on two kinds of characteristics: those linked to the problem and those linked to the computing node that will potentially solve this problem. To measure the node capacities, sensors are used (like the software NWS). These sensors must evaluate periodically or on demand parameters mentioned before. However sensors can be more or less intrusive and their activities can insert mistakes in measurements. Whereas to know problem characteristics, it is possible to use instrumentation code.

Load balancing: this function is associated to performance prediction. In fact, it is the implementation of prediction results. In the next, we will name process one elementary part of the considered calculation or problem. The load balancing is based on processes migration. This migration is the dynamic modification of their physical location. To reduce the network load, we must group on a same node the strong coupled processes (those that make lots of data exchanges to each other). As for the node load, we must move processes of heavily loaded nodes to others less loaded. There is another distinction : load balancing can be **static** and be realized at the beginning of the run or it can be **dynamic** and be occurred during the run. The static load balancing is the subject of lots of research in the hope of optimizing the execution duration. Lots of heuristics [Braun *et al.*, 2001] exist.

Fault tolerance: when one of the grid nodes have a failure, we want to lose a minimum of informations and calculations. For that, two mechanisms are used: checkpoints and rollbacks [Folliot Sens, 1999] [Nguyen-Tuong, 2000]. Checkpoints are storages of the application state (states of processes, memory stacks, ...). They are achieved periodically. They are independent or coordinated with other nodes. Rollback is a jump of the node state to an anterior checkpoint. Then, the rollback takes effect on only one node of the distributed system and can put it in a incoherent global state (causality principle is not respected).

Another distinction and classification criterion of grids is their architecture.

Architecture

Client-Server: a very common architecture for grid computing is Client-Server architecture, also called ASP model (*Application Service Provider*). The servers are computers that have problem solving capabilities. The clients are grid users. They put forward problems to solve. More precisely, it is *agent-client-server* model that is used. We can explain it quickly: each server (that can solve one or more specific problems) registers itself to the agent. The client transmits its problem heading to the agent. The agent goal is to find the appropriate server (this one that knows how to solve the problem and that has memory and processor resources to do it) and to put it in relation with the client. At that moment, the client transmits its problem to the server. The server solves it. It is a centralized architecture, not very adjustable and not very adaptable in which roles of components are frozen.

Peer to Peer (P2P): another architecture exists for grids, particulary for grid database. It is the peer to peer. Each computer of the grid has data and searches other data little by little without centralization or little (we can quote Napster (with a centralized part) and Gnutella). Each computer has the same role as the others. Consequently, this grid is opened and dynamic. This architecture is appealing and a way of research tries to use it for distributed calculations [Ji, 2003]. Several teams are working to build API or middlewares that provide basic services for P2P network [Gong, 2001], [Cappello *et al.*, 2005]. So, each peer has the total control of these local resources, and can get involved or withdraw from the system at any time. However some difficulties inherent in grid computing are staying present (load balancing, task schedule, synchronization) and must be managed in an equitable way by the peers community.
With an overview of the whole peer-to-peer system, we can qualify it of self-organized: it has the dynamic and distributed reconfiguration property in function of the number and the characteristics of nodes.

In computer science, another paradigm uses these characteristics: it is the agent paradigm.

FROM P2P TO MAS

Before to draw a parallel between agents and peers in the next section, we will remind the notions of agent and MultiAgent System.

Agent paradigm

Agent: the concept of agent is clearly defined. An agent is an object (at the computer science sense) that is active and autonomous. [Demazeau, 1996] describes an agent like an real or virtual entity that has an autonomous behavior and evolves in an environment. An agent can perceive its environment, can realize actions on it and can interact with other agents. [Wooldridge Ciancarini,] adds to this definition: an agent is a computing system having capabilities to make actions autonomously and flexibly in an environment. The

flexibility means reactivity, pro-activity and possession of social capabilities. Often, the life cycle of agents follows the sequence of actions [Tisseau, 2001]: *perception* (the agent perceives its immediate environment through specialized sensors), *decision* (it decides what it must do, taking into account its internal state, its sensor values and its intentions), *action* (it acts by modifying its internal state and its immediate environment).

MultiAgents System (MAS): a system made up of several of these autonomous and pro-active entities with a shared environment is called MAS. [Demazeau, 1996] identifies the key-concepts of MAS as being:

- the Agent and especially its knowledge, its goals and its reasoning.
- agent's Environment: all that does not belong to it and on which it can act
- the Interactions between agents: communications, ...
- the Organizations: the dependencies networks between agents and structures built in this way.

Thanks to the autonomy of the agents (in accordance with localization of its resources and its independent behavioral decisions), they can be physically distant to each other and form a distributed MAS.

MAS for grid computing

It is possible to bring closer the grid computing and the MAS. Indeed, it is interesting to notice similarities between the P2P networks and the MAS: geographically distributed, not centralized system, self-organization, autonomy of entities (peer and agent respectively).
It is easy to quote again the MAS characteristics while clarifying this time their significances in the P2P context:

- *Environment*: hardware of the node (processor, memory), other applications running on this computer, the network, all other peers.
- *Interactions*: communication messages, and other data transfers (serialized objects, ...).
- *Organizations*: set of peers collaborating together (during the load balancing for example).

We can quote other characteristics of agents:

- *Collaboration*: static and dynamic load balancing, scheduling , environment consistency.
- *internal state*: available resources, objects of the simulation
- *pro-activity*: each peer participating to the simulation must not be behind the others. At the global point of view, the simulation time should be minimized.

Only few examples combine peer to peer grid computing and MAS: we can quote Tang works [Tang Zhang, 2006]. Her grid computing architecture is a P2P architecture but each node of the grid is composed of 3 agents collaborating together. The MAS is at the node level and not at the grid level.
Thanks to considerations that we detailed, its possible to treat a peer to peer grid computing like a multiagent system.

APPLICATION: DIVA

As indicated in the introduction, our laboratory is equipped with a lot of personal computers linked to each other by a local network. The following observation is true for many of PC, including ours: they are widely underused (during the main time, the use of their processor doesn't exceed 10%). So, our choice is the following: using and adding the available calculation capacities of our computer grid. We will build a *metacomputing* grid by using the Peer to Peer architecture that will be extended with the agent paradigm.

Generalities

The distributed simulation on computer grid is supervised by the DIVA software. This software that is installed on each node of the grid can be viewed like a background task (daemon): this task does nothing when it is not requested. A user can run a simulation from any node. Then, this node has additional responsibilities comparatively to the others (simulation starting, observation tools and user-simulation interactions tools), and conserves also the basic functionalities of each node. These shared functionalities (load balancing, synchronization, spatial and temporal consistency maintenance) allow us to consider that each node has a similar role and is a peer. In addition, as we explained before, these peers are active and pro-active thanks to periodically executed activities (some of these will be detailed after) and to decisions taken by themselves in the aim to cooperate for executing as better as possible the *In Virtuo* experiments. The nodes are actually agents. That justifies the name we give to this software: *Distributed In Virtuo Agents*. This DIVA software is constituted of a Central Decision Making module (CDM) and is complemented by several peripheral modules for the network interface, the observation of resources load, the load balancing, the management of the AReVi simulation (Virtual Reality engine), the periodic save, ... (figure 1).

Collaborations between DIVAs

Some tasks of the grid management are undertaken by DIVA collaborations in our peer-to-peer model. There is no more centralized system. Let us see some examples of collaborations that underline our multiagent view of the grid.

Load balancing: before running distributed simulation, the set of the simulation components (biological interactions, chemical species, ...) are moved onto the set of nodes accepting this simulation. After this static load balancing carried out, the different places of calculations can be changed for optimize the "resources quantity / calculation load" adequacy. This optimization will be carried out dynamically by the DIVA collaboration. Several collaboration types between agents exist. We can cite the famous "Contract Net Protocol" as well

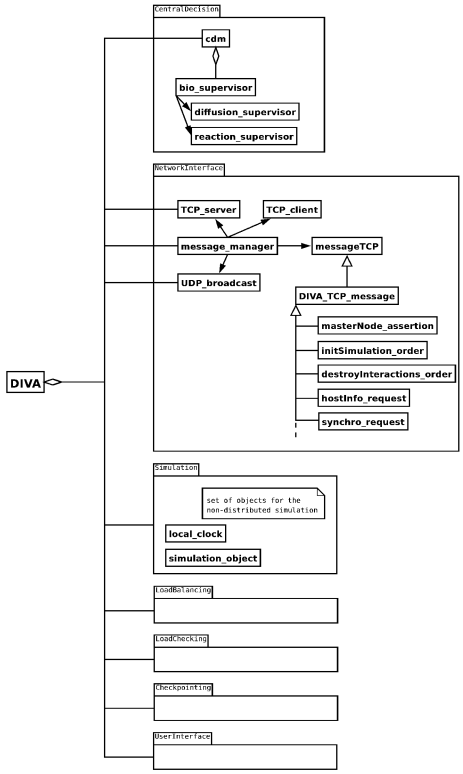


Fig. 1. *Simplified UML class diagram of the DIVA software*

as its improvements and its by-products like auctions [Buyya, 2002].

Time of simulation: There is not global clock in our system. Each node of the grid has its own local clock. So that the whole simulation conserves consistency, these clocks must be the more synchronized as possible. For doing that, each message passing between the DIVAs has a timestamp. During a point to point communication, the receiver DIVA can take the choice to freeze its part of simulation if it is too early compared with the emitter (the threshold is adjustable). So, as all DIVAs have regular bi-directional communications to each other, the local clocks are synchronized two by two and the set of clocks have a very similar time.

Consistency of the environment: in our model, the autonomous entities of the biological simulation are not duplicated. Exclusively the parts of the environment can have several images. It is fundamental that these images of a same environment part don't cause important errors in the simulation. These images must stay in a consistency state. As writing Simon [Keller Simon, 2002], consistency in a virtual world, it is when a scene is identically perceived by all the entities which observe it. So, there is disconsistency when a modification is applied to one of the images and not to the others. These repercussions, or these updates of images are resynchronzations of the images. Each DIVA takes regularly the decision to resynchronize the environment parts that have an image on another

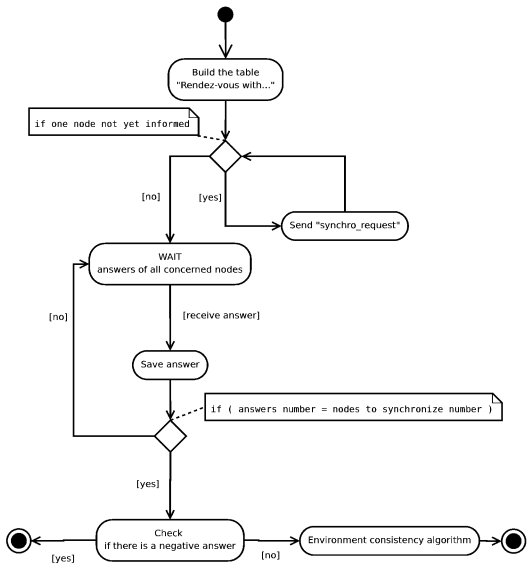


Fig. 2. *UML activities diagram of the "rendezvous" algorithm*

node. For doing that, this DIVA organize a *rendezvous* with the other DIVAs (figure 2). When all concerned DIVAs are at the *rendezvous*, they share the state variations undergone by these elementary volumes: it is the environment consistency algorithm. The new state is assigned to each of the environment parts images. Between two consistency updates, these images are probably non-identical and a "small" error in the simulation is inserted. We must estimate this error and make it negligible compared to the biological model scales. Thus, a consistency update more or less frequent will make a simulation more or less precise but also will bring more or less important loads.

RESULTS

The objective of this distribution software is to able to process the most part or all the biological models that we simulated on one computer. A first example, this one we present here, is the MAPK (mitogen-activated protein kinase) cascade [Kholodenko, 2000].

The MAPK pathway is a series of enzymatic reactions. This pathway is present in all eukaryotic cells and is responsible of lots of control functions for the life cycle of cells. This pathway was modelled in our simulation with the Kholodenko model (figure 3). This model has 10 chemical reactions including a negative feedback. The usual results are the evolutions of the chemical species concentrations. These concentrations oscillate and these oscillations are not deadened during the time. These results are correct in an homogeneous environment. In the present simulation, we build an heterogeneous environment where one chemical species of the pathway is injected in a local area. This species will be diffused in all the environment during the simulation. The figure 4 shows the result of one particular chemical species for two simulations. In the top part of the figure, it is the

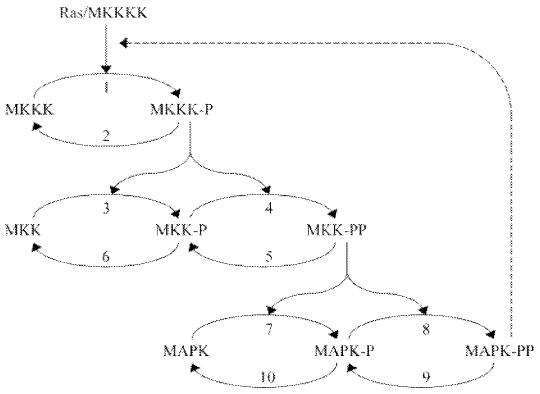


Fig. 3. *MAPK pathway: Kholodenko's model*

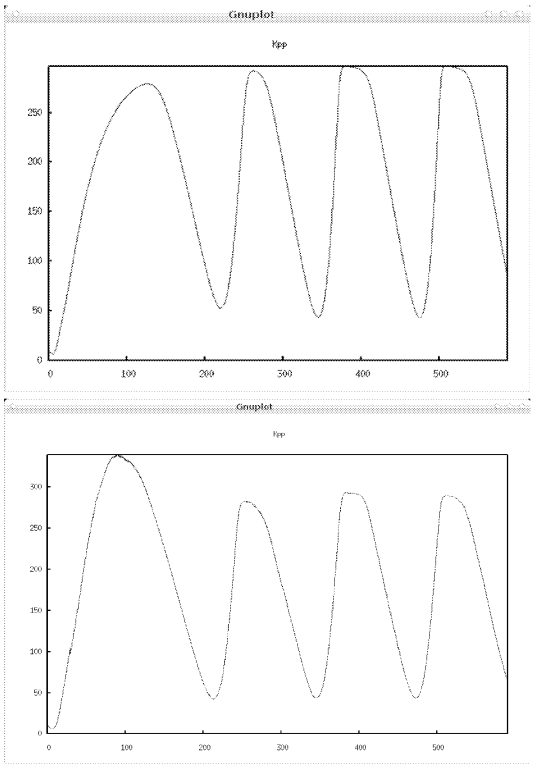


Fig. 4. *Simulation of the MAPK pathway in a heterogeneous environment (with one computer, and with three computers with DIVA software)*

simulation carried out on one computer. In the bottom part, it is the simulation carried out on several computers. The global form of the curve is identical but the amplitude of the first peak is different. This error is caused by some imprecisions of the synchronization.

CONCLUSION AND PERSPECTIVES

We presented here the themes of our research works i.e. how to accumulate the resources of computers to carry out *In Virtuo* experiments at large scale. To be able to answer it, we gave an overview of the grid computing first. This state of the art allowed us to make choices for the development of this project. Thus, we use the PCs of our laboratory local network to build a grid. The main point that we defend is the architecture of each node of this grid. In effect, we want a minimal centralization in this architecture: all of the nodes have similar roles; they are peers. We extend this reasoning more far by giving an autonomy, a decisional capability and a pro-activity to these peers. Each node of the grid is henceforth considered to be an agent. The software that we are developing to make this concept a reality is called DIVA.

We begin the first simulations with this software. We hope to be able to carry out the totality of the already existing simulations with this grid soon. Next, we will build models which require lots of calculation resources and for which only one PC can't simulate in real time.

REFERENCES

[Braun *et al.*, 2001] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen, R. F. Freund. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, 61(6):810–837, 2001.

[Buyya, 2002] R. Buyya. *Economic-based Distributed Resource Management and Scheduling for Grid Computing.* PhD thesis, Monash University, Melbourne, Australia, April 12 2002.

[Cappello *et al.*, 2005] F. Cappello, S. Djilali, G. Fedak, T. Herault, F. Magniette, V. Néri, O. Lodygensky. Computing on large-scale distributed systems: Xtrem web architecture, programming models, security, tests and convergence with grid. *Future Gener. Comput. Syst.*, 21(3):417–437, 2005.

[Demazeau, 1996] Y. Demazeau. Vowels. *Invited lecture, 1st Ibero-American Workshop on Distributed AI and Multi-Agent Systems (IWDAIMAS 96)*, Mexico, 1996.

[Folliot Sens, 1999] B. Folliot P. Sens. Load sharing and fault tolerance manager. *High Performance Cluster Computing: Architectures and Systems*, 1(22), March 1999.

[Foster *et al.*, 2001] I. Foster, C. Kesselman, S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International J. Supercomputer Applications*, 15(3), 2001.

[Gong, 2001] L. Gong. JXTA: A network programming environment. *IEEE Internet Computing*, 5(3):88–95, May-June 2001.

[Ji, 2003] L. Ji. Computation in peer-to-peer networks. *Proceedings of the 2002-2003 Grad Symposium*, 2003.

[Keller Simon, 2002] J. Keller G. Simon. Toward a peer-to-peer shared virtual reality. *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, 695–700, Washington, DC, USA, 2002. IEEE Computer Society.

[Kholodenko, 2000] B.N. Kholodenko. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascade. *European Journal of Biochemistry*, 267(6):1583–1588, March 2000.

[Nguyen-Tuong, 2000] A. Nguyen-Tuong. *Integrating fault-tolerance techniques in grid applications.* PhD thesis, Faculty of the School of Engineering and Applied Science at the University of Virginia, Charlottesville, Virginia, USA, August 2000. Adviser-Andrew S. Grimshaw.

[Tang Zhang, 2006] J. Tang M. Zhang. An agent-based peer-to-peer grid computing architecture: convergence of grid and peer-to-peer computing. *ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research*, 33–39, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[Tisseau, 2001] J. Tisseau. *Virtual Reality* –in virtuo *autonomy*–. Accredition to direct research, field computer science, University of Rennes I, Rennes (France), December 2001.

[Wooldridge Ciancarini,] M. Wooldridge P. Ciancarini. Agent-Oriented Software Engineering: The State of the Art. *First Int. Workshop on Agent-Oriented Software Engineering.*

Multi-Agent Systems and Negotiation in a Group Forming Scenario

Ingo Stengel
Udo Bleimann
Mathias Kreider
Martin Kanold
IGDV – Centre for Advanced Learning,
Media and Simulation
Hochschule Darmstadt,
Schöfferstr. 1
64295 Darmstadt, Germany
E-mail: ingo.stengel@h-da.de

Jeanne Stynes
Institute of Technology Cork
Computing Dept.
Rossa Avenue,
Bishopstown, Cork
County Cork, Ireland
E-mail: jstynes@cit.ie

KEYWORDS

Education, optimization, combined simulation, AI-supported simulation, dynamic model

ABSTRACT

An approach for group forming using self-oriented agents in combination with a central component is presented. The basic architecture extends Zhang's architecture (Zhang et al. 2003). It introduces a negotiation model and adds a solution for the communication of representative agents. Further details about the implementation are given. First results are analyzed and discussed.

MOTIVATION

Today the use of agent technology is advantageous in applications where interests of persons are represented, where agents act autonomously on behalf of these persons. They are particularly used in applications where individual views and goals on common problems are solved. To achieve optimal results they are using, among others, negotiation and coalition forming strategies (Vauvert et al. 2001).

PROBLEM DESCRIPTION

The research is focused on difficulties that appear when negotiation and group building processes in multi-agent systems are employed to deal with complex issues. For a better understanding, the approach is used in a scenario where agents support students to form a learning group.

In this context two interests and views are evident:

- the "personal" view of self-driven agents - representing the interest of the students,
- the "central" view of the system – interested in placing all the students in a number of learning groups.

Our intention is to extend an existing FIPA (Foundation for Intelligent Physical Agents) conform platform so that it is able to solve our group forming problem. The main aim is to investigate the behavior of a real system and to optimize it, until a solution has been found.

MODEL

We define an approach to form coalitions, preserving an individual vision of the world for the single agents, but also take into account a central perspective. The following solution is using self-interested and cooperating agents, as described by (Ferber, 1999), and is able to form coalitions and to negotiate in a multi-layer architecture.

Many approaches to model agents having "human" qualities exist, e.g. the BDI Agent-Model (Georgeff, 1989), the PECS Model (Schmidt,2000), etc. Nevertheless, as stated above, solutions must be adapted to the problems. The parameters used are oriented respectively on the problems to be solved and on the goals to be reached.

The virtual university scenario introduces, amongst others, the following assumption: all competing students for a learning group membership are equal. This means that all agents start with identical knowledge. Only the "personal" parameters are diverse and relevant.

Every student i has a number of n attributes $attrib_i$, e.g. years of study, a self-evaluated number identifying his level, etc. Additionally, every student is looking for learning comrades that have certain attributes $search_i$. Any of these attributes have a certain priority $prio_i$.

The model is generic in its search. The types of attributes are depending on the model used and on the goal to be reached. So if the PECS model (Schmidt,2000) is used, attributes referring to intelligence, knowledge and social abilities should be common. When the goal to be reached is the formation of a learning group attributes characterizing a learning preference model are more specific, as in the learning preference model by (Roell,2005).

The attributes of agents can be very different. This makes group forming and the negotiation a multidimensional process. To improve the ascertaining of solutions, priorities were added to the attributes of an agent. Using a weighted Euclidian distance function (1), this will give a rating to the offer an agent is receiving.

$$d = \sqrt{\sum_{i=0}^{k} prio_i \times (search_i - attr_i)^2} \qquad (1)$$

The distance *d* between two agents [1] shows how well the request of one agent is fitting the offer of the other agent. In addition the resulting values are normed, so that they are comparable.
To prove that this "reduction" of dimensions is working, it has been simulated with uniformly distributed input values.

The red points in fig. 1 show the distribution of the values of the attributes of the agents, while the blue vector contains the values of the solution the agent is looking for. The green vectors are two best solutions that have been selected due to the weighted Euclidian distance function.

This distance is used in the negotiation process to determine the adjacency of requests ($search_i$) and offer ($attrib_i$) between agents in our model.

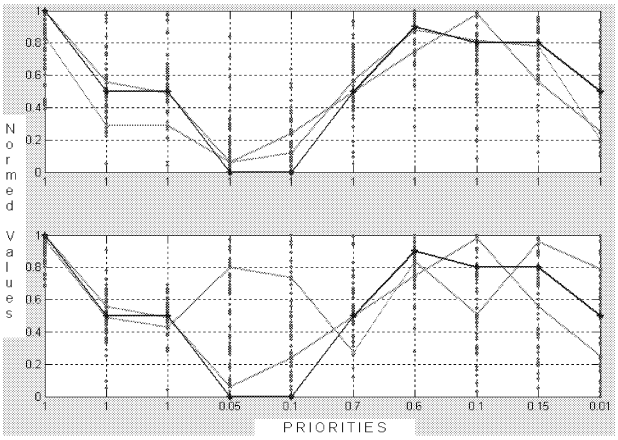


Figure 1: MATLAB Simulation of the weighted Euclidian distance function

A lot of research has been done in the area of cooperation models e.g. by (Axelrod, 1997).

Negotiation strategies must be adaptable and able to use multi-dimensional attributes. In this context agents must use the same ontology.

In our model the negotiation process takes place in five steps as known in usual marketplaces. The contract state is reached after the inform message has been received by one of the negotiating agents, see fig. 2.

If during the negotiation the distance is below a certain threshold (desperation threshold), then a group consisting of these two agents can be formed. Since the system is fully dynamic, the group will have to negotiate further with other agents. Therefore a representative agent (Yamamoto, 2001) is generated. The role of this agent is to represent the interests of the group. But since the goals are not identical, new goals of the group must be computed.

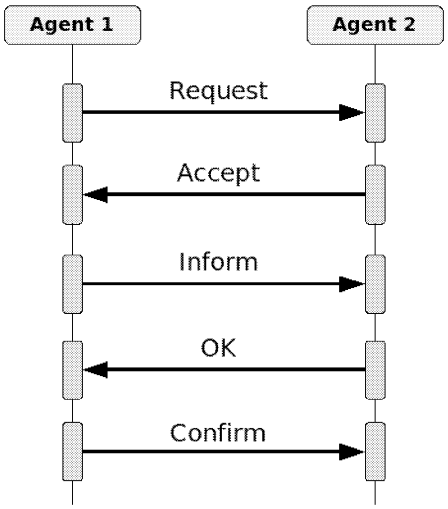


Figure 2: Negotiation between 2 agents

As well known from other approaches (xx) the values of the attributes are averaged (2)(3).

$$attr_j = \frac{\sum_{i=1}^{m} attr_{i,j}}{m} \qquad (2)$$

$$prio_j = \frac{\sum_{i=1}^{m} prio_{i,j}}{m} \qquad (3)$$

m is the number of agents in the group.

The search vector is calculated taking into account the goals of every agent in the group (4)(5).

$$sumprio_j = \sum_{i=1}^{m} prio_{i,j} \qquad (4)$$

$$search_j = \frac{\sum_{i=1}^{m} prio_{i,j} \times search_{i,j}}{sumprio_j} \qquad (5)$$

In figure 3 the values of the priorities and the search vector are shown in green, while the values of the representative's search vector are shown in yellow.

Groups can be formed by two agents, by an agent and a group or by two groups.

When an agent and a representative form a group then the representative will recalculate (2)(3)(4)(5) by taking into account the values of the attributes of the agent that joined the group. In the case of two representatives deciding to merge their groups, one will be terminated, while the other will recalculate its values depending on the participants of the group.

To avoid situations in which agents in groups negotiate with other agents a redirect mechanism has been introduced. This redirects any message send to an agent that is part of a group to its representative. After this redirect, the other agent can negotiate directly with the representative of said group.

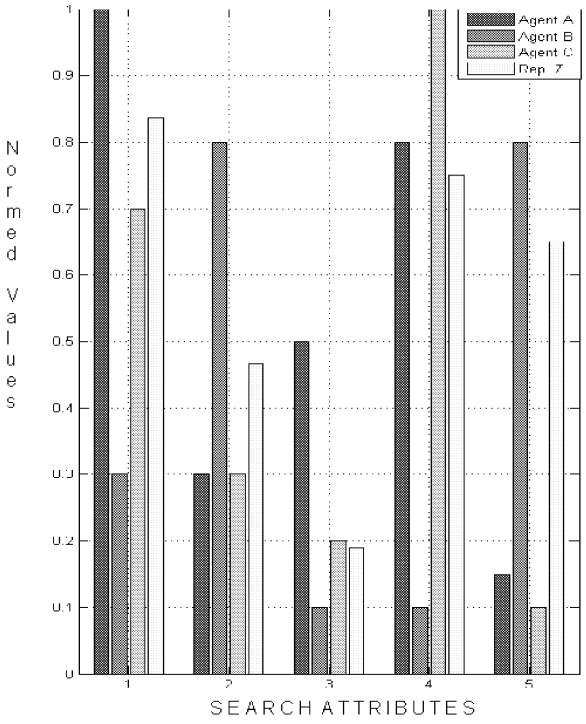


Figure 3: Example of a group with calculated search vector

As a criterion for a qualitative comparison of different solutions, the *harmony-value of a group* $group_g$ has been introduced. This value characterizes the quality of the group. It shows how well the members of the group are fitting together (6). Where m is the number of agents in a group and g is the number of the group. $d_{i,j}$ represents the distance between agent i and agent j of a group.

$$group_g = \frac{\sqrt{\sum_{i=1}^{m}\sum_{i=1}^{m-1} d_{ij}^2}}{m\times(m-1)} \qquad (6)$$

In this approach, the situation of agents leaving a group has been avoided since this makes dispersing coalitions more probable, like stated in (Ming et al. 2005). When one agent leaves, then it is very probable that others will leave as well.

ARCHITECTURE

In the early years (Shehory et al., 1998) defines a completely distributed approach that uses an algorithm to calculate all possible coalitions for every agent.
Looking for a distributed solution (Sandholm et al., 1999) suggests the distribution of the search space of agents. Unfortunately he still rely's on centralized information. Further he suggests the reduction of the complexity by reducing the number of agents per coalition.

Starting from the usual one level negotiation framework, e.g. (Tu *et al.*, 1999), the concept has been split up into a two-level negotiation framework in (Zhang *et. al.,* 2003) and (Zhang *et al.*, 2006) bringing two existing frameworks – the motivational quantities framework (MQ) and the Task analysis, Environment Modeling and Simulation framework (TAEMS) - together. Based on this architecture which is combining a "neutral central solver" with a distributed agent negotiation model, (Lau *et. al.*, 2004) improves the two layer coalition forming model.

Our architecture, supporting multiple negotiations, is based on the two-layer multi-agent framework used in (Zhang *et al.*, 2003). It includes a central component (CC) that is concentrating on the quality of the solution, a layer that is enabling negotiation between agents and which additionally is reducing the quantity of agents involved.

The CC is working like a broker (Stengel et al. 2006). It offers any agent references of possible agent partners. Hopefully, with this information, every agent will be able to form coalitions and solve problems by negotiating with other coalitions and agents.

Before starting negotiations, the CC is calculating the distances between the agents. Then a personalized list of possible n partners is submitted at initialization time to each agent.

To be able to support multiple negotiations, a structure including multiple negotiation units has been defined. Every agent controls his sequence of negotiation units. See figure 4.

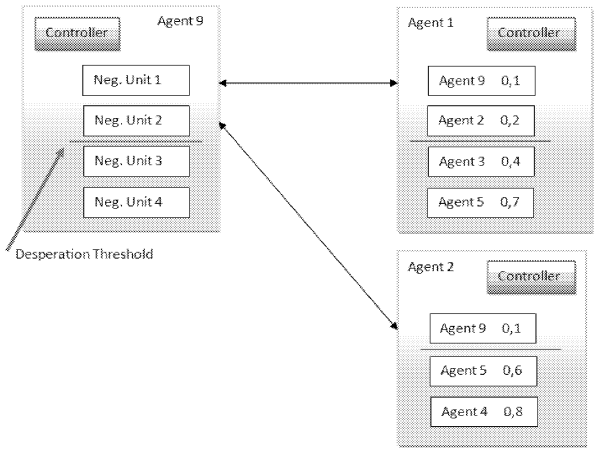


Figure 4: Negotiation units controlled by agents

The negotiation unit is implemented similar to an automaton, see figure 5; additionally the sequence of them can be automatically sorted. To reduce the amount of complexity, the desperation threshold has been defined. Any negotiation unit able to negotiate with agents in a distance below the threshold will be activated.

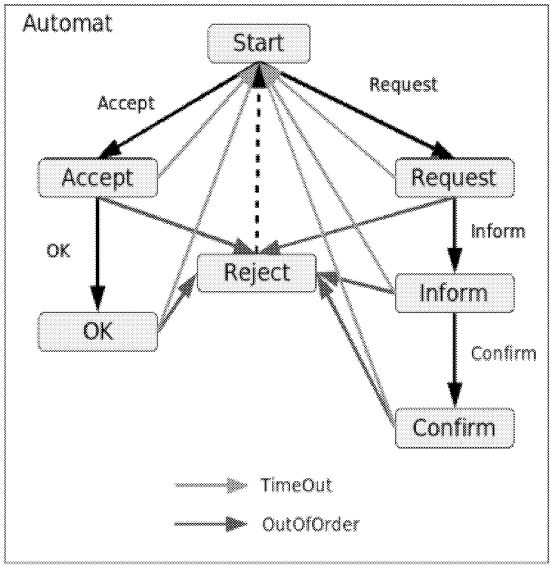


Figure 5: Basic negotiation unit (automaton)

The others in the list will be inactive until the threshold will reach them. If no possible partner is found, the threshold is increased. Now the list of possible agents to negotiate with is extended, increasing the possibility of finding the right fit.

After a certain amount of time, all agents stop their activities and deliver their information to the CC. Now the results are analyzed. If all agents have not been placed in groups, then the CC will start a central algorithm to place the remaining agents.

IMPLEMENTATION

Starting with JADE - a FIPA compliant multi agent system - as the basis for our implementation, the two layer negotiation architecture is constructed on top of it. The group forming system consists of four main java packages: agent, behavior, decision and ontology. These have been implemented during a period of over one year.

EXPERIENCES

Most of the occurring problems were deadlock situations. These problems have been solved by introducing a timestamp mechanism and several safeguards against the formation of overlapping groups.

The behavior of the system is not necessarily simply the sum of the agents' behaviors. That could result in unpredictable states (Remondino, 2004). The only approach to this problem is to statistically analyze multiple runs with varying initial conditions (Axtell, 2000).

Figure 6 shows the amount of agents in groups over time during multiple runs without the optimizing activity of the CC. As can be seen, the solutions are not identical, since the agents are in competition and the sequence of execution is depending on several factors, e.g. scheduling of the OS, scheduling of the MAS, scheduling the sending and receiving of the messages.

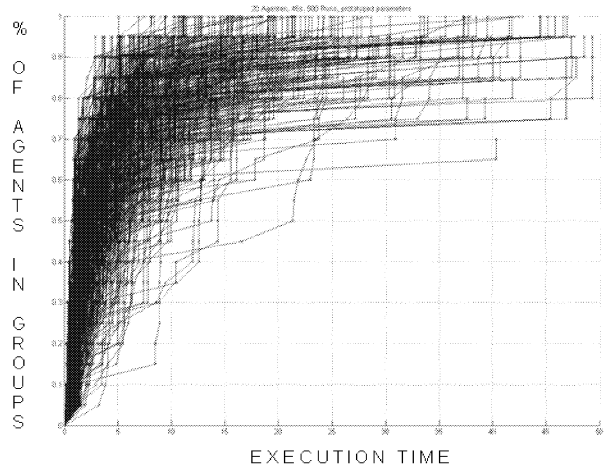


Figure 6: Multiple run: 20 agents, 45 sec, 500 runs, no CC

However the best solutions are found in the first 15-20 sec. After this time the desperation threshold is increased. The solutions found during this time can diverge from the desired solution. The harmony values of the groups set up during this time are not as good as the harmony values of the ones found in the first 20 second.

Determining the optimization point P (t1,X) for the switch between the agent based approach to a central approach.

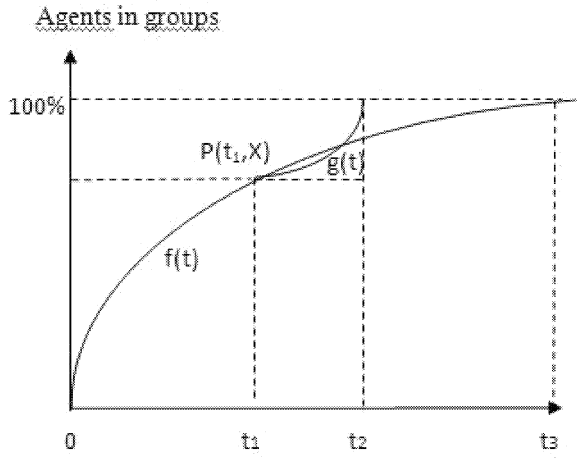


Figure 7: Determining the optimizing point P for the method switch

The function f(t) represents the evolution of a solution using exclusively multi agent negotiation. The function g(t) represents the progress of solutions over time by the central component (central solution). t_1 is the time when it makes sense to switch between the two methods. The selection of this point $P(t_1,X)$ makes sense as soon as the difference between the termination time t_3 of the agent based approach and the termination time t_2 of the composed approach (agent based and later central) $(t_3-t_2) > k$, where k is a predefined threshold, that has to be evaluating set as a result of the analysis of multiple runs.

CONCLUSIONS

In this approach we extend the design of the architecture for negotiation by linked negotiation units controlled by an agent. Additionally the architecture is based on the FIPA standard architecture.

A „social" strategy in problem-solving is used by combing the central problem solving with the agents self-oriented approach and with a central coordination approach (periodical hints to agents).

The negotiation strategy used is able to support multi-attribute oriented group forming.

The communication strategies during negotiation in groups have been extended by defining redirects to representatives.

In the near future, the comparison to a normal multi agent system will be done by analyzing a lot of multiple runs on a dedicated server.

REFERENCES

Axelrod, R. 1997. "The complexity of cooperation: agentbased models of competition and collaboration", Princeton, NJ: Princeton University Press, 1997.

Axtell, R. 2000. "Why Agents? On the varied motivations for agent computing in the social sciences", Center on Social and Economic Dynamics, Working Paper No. 17

Ferber, J. 1999. "Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence", Addison-Wesley, 1999, ISBN: 0-201-36048-9

Lau, H.C., Zhang, L. 2004. "A Two-Level Framework for Coalition Formation via Optimization and Agent Negotiation". IAT 2004: pages 441- 445

Georgeff, M.P., Ingrand, F.F. 1989. "Decision –Making in an Embedded Reasoning System", Proc. Of the 11th International Joint Conference on Artificial Intelligence, Detroid, August, 1989.

Remondino, M., Cappellini, A. 2004. "Minority game with communication: an agent based model", in *Proceedings16th European Simulation Symposium, György Lipovszki, István Molnár* © SCS Press, 2004.

Roell, F.J. 2005. "Der Lernwegweiser im Internet", http://www.lerno.de, 2005, last accessed 7.04.2006.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohme, F. 1999. "Coalition Structure Generation with Worst Case Guarantees", Artificial Intelligence 111, pages 209-238.

Schmidt, B. 2000. "Modelling of Human Behaviour, The PECS Reference Model", SCS Europe, Ghent, 2000.

Shehory, O., Kraus, S. 1998. "Methods for Task Allocation via Agent Coalition Formation", Elsevier Science.

Stengel, I., Bleimann, U., Stynes, J. 2006"An Approach to Forming Coalitions and Negotiating in Multi-Agent Environments", *Proceedings of the International Networking Conference 2006*, University of Plymouth, UK.

Tu, T., Seebode, C., Lamersdorf, W. 1999. "A Dynamic Negotiation Framework for Mobile Agents" in *Proceedings of the 3. Intl. Symposium on Mobile Agents*, MA'99, Palm Springs, California IEEE Computer Society Press.

Vauvert, G. and Seghrouhni, A. 2001. "Coalition Formation among Strong Autonomous and Weak Rational Agents", in *Proc. 10th European Workshop Modelling Autonomous Agents in a Multi-agent World (MAAMAW)*, Springer Verlag, Heidelberg, 2001.

Xie, M.,Chen, J. 2005. "Studies on Horizontal Cooperation among Homogenous Retailers Based on Multi-Agent Simulation", in *Proceedings of the 3rd IEEE International Conference on Industrial Informatics* (INDIN), 2005.

Yamamoto, J. and Sycara, K. 2001. "A Stable and Efficient Buyer Coalition Formation Scheme for E-Marketplaces", in *Proc. of AGENTS'01*, Montreal, ACM, 2001.

Zhang, X., Lesser, V., Wagner, T. 2003. "A Two-Level Negotiation Framework for Complex Negotiations". IAT 2003: pages 311-317.

Zhang, X., Lesser, V., Wagner, T. 2006. "Integrative negotiation among agents situated in organizations". IEEE Transactions on Systems, Man, and Cybernetics, Part C 36(1): 19-30.

BIBLIOGRAPHY

Ingo Stengel is PhD Student at the Computing Department at the Cork Institute of Technology (CIT) Ireland; co-founder and Executive Director of the igdv - Centre for Advanced Learning, Media and Simulation in Darmstadt. His research interests are in the area of multi agent systems, simulation software and advanced learning.

Udo Bleimann graduated at the University of Frankfurt, since 1985 he is Professor for computer science at the University of Applied Sciences Darmstadt; founder and director of the igdv – Centre for Advanced Learning, Media and Simulation, director of the Institute for Applied Informatics Darmstadt (aida). His research interests are in the area of advanced learning, telecommunication and distributed systems.

Mathias Kreider is a researcher in the Multi Agent System project at the igdv - Centre for Advanced Learning, Media and Simulation, Darmstadt, Germany

Martin Kanold graduated at the Computer Science Department of the University of Applied Sciences Darmstadt. Now he is a researcher at Siemens, Germany.

Jeanne Stynes is senior lecturer at computing department of the Cork Institute of Technology, Ireland. Her research interests are in the area of distributed systems.

MULTI-AGENT SIMULATION OF AN EDUCATIONAL COLLABORATIVE WEB SYSTEM

Álvaro Barbero, Mario Salvador González-Rodríguez, Juan de Lara, Manuel Alfonseca
Escuela Politécnica Superior – Universidad Autónoma de Madrid (Spain)
{Alvaro.Barbero, Mar.Gonzalez03}@estudiante.uam.es, {Juan.deLara, Manuel.Alfonseca}@uam.es

KEYWORDS: Agent simulation, education, SWARM.

ABSTRACT

This paper describes the simulation of the activities of a group of students who are using a simulated collaborative educational system, based on a real system (called *KnowCat*) used by real students in several courses in our department. The simulation has been implemented as a multi-agent system, where both the students and the work they perform are represented by interacting agents. The objective of the simulation is to provide a means to analyze different situations and to study global system behaviour and effects such as the quality of the most voted documents, or the overall learning progress and motivation of students.

1. INTRODUCTION

The Internet has become an essential tool for many kinds of service providers, which have adapted their business organization to benefit from this media. In particular, educational courses have evolved from traditional lessons in the classroom to on-line systems, which offer rich interaction possibilities (both synchronous and asynchronous) between learners, and between learners and teachers. These systems allow a more flexible learning process, by permitting the access to educational contents anywhere, anytime.

On-line educational systems are less guided than traditional learning, promoting exploration and a more active role of the student. In this way, in some systems the student is responsible on his own for choosing the learning path among the available contents. In an educational context, groupware [G94] and collaborative systems [AC99], promote team-work and problem solving, for example by making the students work together by creating parts of the educational materials, or supporting joint activities to evaluate, classify or modify such materials.

Evaluating and testing this kind of web based systems is difficult, as it requires many users interacting concurrently. In educational systems, sometimes it is beneficial to study and analyse the effects that can be produced due to the different kinds of interaction: user-system, and user-user (directly or through the system). A model of these interactions prior to implementation would enable to better design and organize the educational system so as to maximize its usefulness.

Thus, we propose in this paper the use of modelling and simulation for the study of educational web systems. In particular, we advocate the use of agent-based modelling and simulation [JSW98] for this purpose. This is a powerful and natural way to carry out complex simulation experiments where many autonomous and interacting entities take part. The key abstraction in this methodology is the autonomous agent. According to [JSW98], an agent is "*a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives*".

One of the most interesting things to study in this kind of systems is emergence [GT99]. This phenomenon occurs when interactions in a large population of objects at one level give rise to different types of phenomena at another level. In our case, we study the global system efficiency (e.g. student satisfaction, educational material quality, increase of motivation and learning retention) from local student interactions. In particular, in this paper we have built a model inspired by the KnowCat [AC99] system using the Swarm [MBLA96] libraries for multi-agent development. The work is aimed at providing a framework for experimenting different situations and assumptions about student behaviours. We believe that modelling and simulation can be a valuable means for decision support for this kind of systems, and may help in (re-)organizing the real system for maximizing the results.

The rest of the paper is organized as follows. Section 2 gives an overview of the KnowCat system. Section 3 presents the simulation model. Section 4 discusses the implementation and shows some experiments. Section 5 compares with related research and finally section 6 ends with the conclusions and future work.

2. THE *KNOWCAT* SYSTEM

KnowCat is a web-based collaborative system [AC99] which achieves the self-organisation of a set of documents, created and provided by a group of users, in such a way that the system is able to determine the quality of each document as a function of evaluations by the users themselves and their behaviour. The system uses these data to provide naive users with suggestions on the documents that they should study. This system has been used in practice in several courses offered in our department, where the students generate sets of documents about the subject matter of the course in such a way that those with the best quality can be used by the same students participating in the experiment, or those who come in subsequent terms.

In the following, we describe the most relevant features of the *KnowCat* system, of interest in the frame of the experiments described in this paper. They occur as three different phases, which correspond to different student interactions. In the first phase (***creation***), teachers divide the subject matter into a set of *themes* and assign each student one or several of them. The student must build a document for each assigned theme, describing the associated concepts and ideas, using notes taken in the classes, the recommended bibliography, or any other appropriate related source. In this phase, the only interaction of the students with the *KnowCat* system corresponds to the actual delivery of the documents.

In the second phase (***voting***), each student is assigned a number of votes and a different set of themes to be revised. They must then read the documents provided by other students and assign their votes to those documents considered to be the best.

Finally, in the third phase (***reading***), the system classifies the documents according to the votes they have received. The best ones are then recommended to students asking for information about a given theme as support material for the subject matter under study.

When the system is used according to the preceding phases, a lot of information is generated in the form of documents whose quality is not tested by experts in the subject matter, but by the students themselves through the assignment of their votes. However, the votes may not be objective, because the students tend to give their votes to documents written by their acquaintances and friends, or to those easier to read (or shorter ones) but which may not provide a general review of the theme. It may well happen that the most voted documents are not really the best, and the objectives of the use of the system would not be reached.

The application described in this paper simulates the *KnowCat* system and its use by the students, so as to obtain information and perform experiments on the effects of the behaviour of the students on the quality of the generated documents, and the knowledge the students would be able to get from their use as support material.

3. SIMULATION MODEL

First, we start describing the structure of the model. Figure 1 shows a class diagram of the simulation model. Three classes of agents are defined in the simulated system: students, documents, and the *KnowCat* system itself, which envelopes all the others. Documents are purely reactive agents, since they do not perform any action on the other agents. On the other hand, both students and the simulated *KnowCat* system are active agents, as they perform actions on other agents by their own initiative, although they are also capable of receiving external influences.

The *KnowCat* agent marks the start and the end of each of the three phases described in the preceding section, and assigns tasks to the students. The simulation is performed in a time frame where the unit is the week. During each week, the students will try to solve the tasks they have been assigned by the system. Depending on the phase of the work, they can also perform voluntary activities by their own initiative.

The student dynamics is inspired in part in a similar system built by [K05], where the students are assigned a level of motivation which increases when they successfully perform the assigned tasks and decreases when they fail. If their motivation reaches zero, the students stop using the system. In addition to this, each student is provided with a list of knowledge which changes while they use the system. The set of knowledge reached by the students at the end of the simulation is a good indicator of the success or failure of the system.

Compared to the real system, the model applies a certain number of simplifications. During the creation phase, to allow the students finding information about the themes they have been assigned, several documents are initially provided by the teacher. The student should read them before creating the documents assigned to them. After the first phase is finished, these starting documents are removed from the system. Also, during the voting phase, no themes are assigned to the students, who must chose by themselves the documents they want to read and vote.

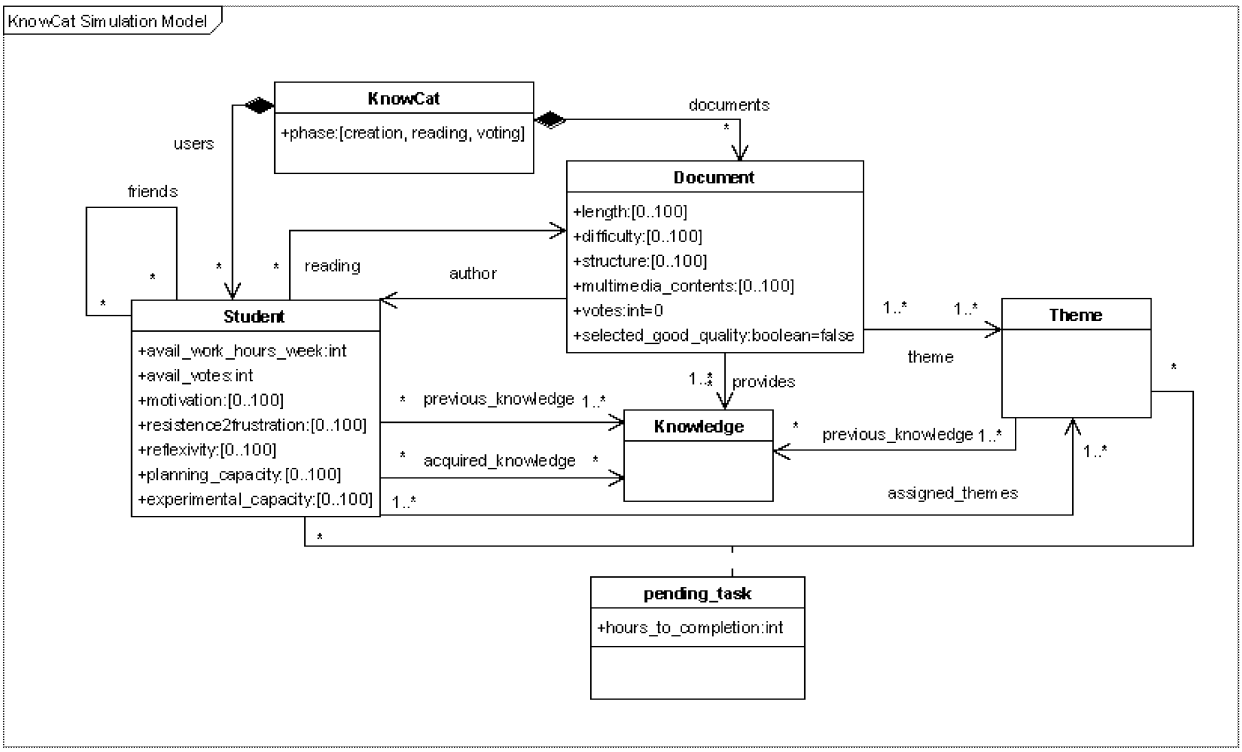


Figure 1: Class diagram of the simulation model.

Two additional entities are defined: knowledge, and themes. Knowledge is divided in two kinds: previous knowledge, which the student possesses as the consequence of previous studies, and acquired knowledge, which corresponds to the current course. The course is assumed to be divided into themes, each of which requires a set of previous knowledge to be understood. A student acquires knowledge about a theme by reading one of its associated documents. The assignment of knowledge to themes is automatically performed by the *KnowCat* agent.

A *document* is an agent containing the following attributes: its length (an integer between 1 and 100), difficulty (a percentage), degree of structure (a percentage), multimedia contents (a percentage), the number of votes obtained by the document (initially 0), and a switch indicating whether the document has been selected by the system as a good quality document. A document is related to the assigned theme, to the required previous knowledge (which is the same for all the documents associated with the same theme) and to the list of knowledge which can be gained by studying the document, a subset of the knowledge associated to the theme. A good quality document should ideally provide all the knowledge associated to its theme. A document is also related to the student who authored the document.

A *student* is an agent with three types of attributes:

- Character definition: resistance to frustration, degree of reflexivity, experimental and planning capabilities.
- Dynamic properties: motivation, affected by the resistance to frustration; number of hours available per week, which varies as a function of the student motivation and can increase or decrease as a consequence of the student behaviour during the current week.
- Accounting information: number of votes which the student can assign during the voting phase (a maximum value, the students may assign a smaller number of votes depending on their motivation); their pending tasks and the number of hours needed to complete them (relation *pending_task*).

A relation links students to their previous knowledge, which assumedly has been acquired in previous courses. Its value varies between one third and one hundred percent of the maximum knowledge which the student can have in the system. This includes the newly acquired knowledge, which the student gets during the simulation, which has an association to the document being read, the assigned themes, and his friends.

Next we describe the dynamics of the model, by presenting the inter-agent interactions and the student behaviour.

Student-document interaction. When the students read a document, the system quantifies their *degree of understanding* as a function of the following data:

- A comparison between the current knowledge of the student versus the list of required knowledge associated to the document. This factor has as much weight as the next two together.
- A function of the student reflexivity, planning capacity and motivation versus the length, difficulty and structure of the document.
- A comparison of the experimental capability of the student versus the multimedia contents of the document.

Every time a student reads one document, a part of the available hours for the current week is expended. The motivation of the student is also adjusted as a function of the understanding obtained from every document read.

Student-KnowCat interaction. The student interacts with the *KnowCat* system in the following ways:

By being assigned themes and the documents which must be read in the first phase.

By uploading documents to the system when they have been created.

By reading and voting documents created by other students.

By reading (optionally, if they have sufficient motivation) additional system documents.

Student behaviour. Student behaviour depends on the phase the system is:

Creation phase: the students read the initial documents associated to the themes they have been assigned by the *KnowCat* system and create new documents related to those themes. Each task in this phase may take several weeks, as a function of the degree of understanding obtained by the students from the initial documents and their motivation.

Voting phase: The students read documents generated by other students. If a minimum degree of understanding is obtained, the document receives one vote. Otherwise, the document is voted with a certain probability if the author belongs to the list of friends of the voting student.

Reading phase: The students may read additional documents in the system, to get extra knowledge. If the students are sufficiently motivated, they will participate in this phase, otherwise they will do it with a 50% probability. If a document has been marked with good quality, it is always read, otherwise it will be read with a 50% probability.

4. IMPLEMENTATION AND EXPERIMENTS

To simulate the *Knowcat* system, we used the Swarm multi-agent simulation library [MBLA96], which is based on ObjectiveC and Java and provides facilities to simulate sets of agents, each of which may recursively contain other agents. The simulation can be dynamically controlled, in such a way that the state of the agents can be modified or tested at any point, making it possible to extract measurements in real time. The three kinds of agents (student, Knowcat and Document) are classes that inherit from the swarm class *SwarmImpl*. As classes student and Knowcat are *active* agents, they have an associated action table (swarm class *Schedule*), so that they can autonomously perform these actions.

The previous interface allows configuring the initial parameters of the student population (i.e. its number and characteristics), the Knowcat system (e.g. the number of themes, the initial documents) and the environment (e.g. influence of friendship in the voting process).

Four different simulation experiments have been performed. Figures 2, 3 and 4 show the result of the *first set of experiments*, where the default values were used for all parameters. Motivation starts slightly above 45 and increases along the simulation until reaching a value of about 60. The number of documents generated per student is below 1, even though 2 had been requested (84 vs. 200). The number of votes assigned was also very small (156 vs. 500 requested, 5 per student). The amount of knowledge acquired by the students was slightly better than 20% (40 pieces of knowledge). This experiment confirms that students starting with a low motivation level do not reach satisfactory results.

The *second set of experiments* was performed increasing the student motivation. When the initial motivation average is set to 70 with a deviation equal to 10, better results are obtained. During the simulation, the average motivation increases to 90. 93% of expected documents are generated (186 vs. 200), and 70% of the expected votes are assigned (350 vs. 500). The average number of pieces of knowledge acquired by the students increases to near 60.

In the *third set of experiments*, the motivation of the students is the same as in the second experiment, but the average difficulty of the initial documents is increased to 90, with a 10 deviation. In this case, the motivation does not increase during the creation phase, although it improves during the voting and reading phases. The number of generated documents decreases by 17 (9%) while the votes assigned by the students also get below the 350 of the preceding experiment.

The *fourth set of experiments* maintained the same parameter values as the third, except that the average frustration resistance is set to 10 (rather than 50, which is the default) with a 10 deviation. In this case, the students cannot increase their motivation, which gives rise to significantly smaller numbers of generated documents and votes.

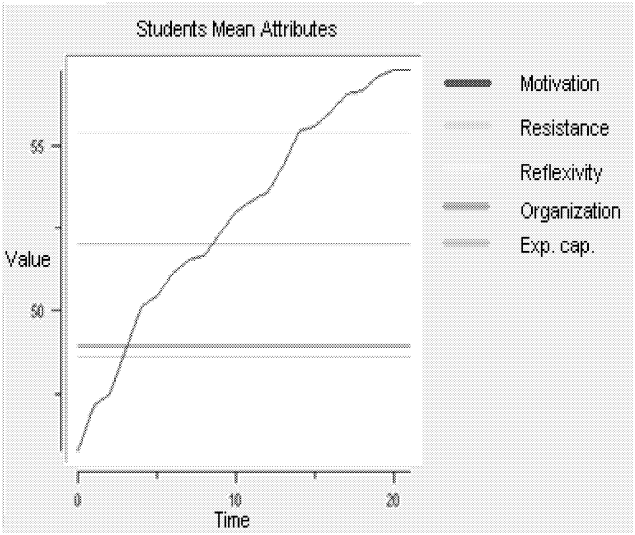


Figure 2: Average student attributes (1st experiment). Colors respectively represent average motivation, resistance to frustration, reflexivity, organization and experimental capability.

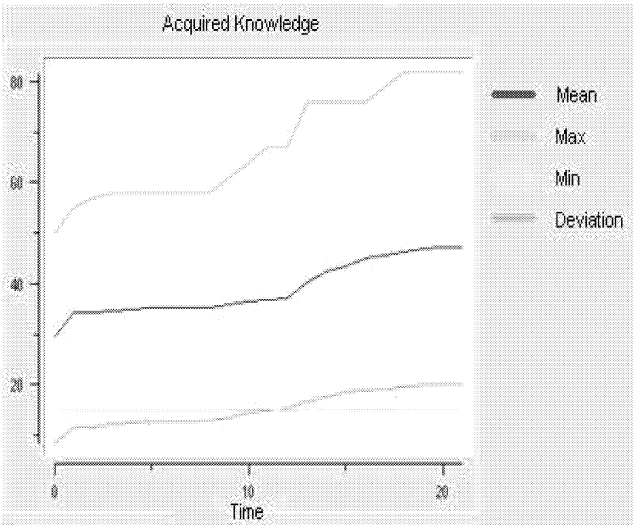


Figure 3: Acquired knowledge (1st experiment).

We observe that the average motivation of the students tends to increase with the use of the system. This increase depends on the initial attributes of the students and the difficulty of the assigned tasks. In the case of high levels of motivation and resistance to frustration, combined with a low level of task difficulty, the results obtained by the students interacting with the system are satisfactory.

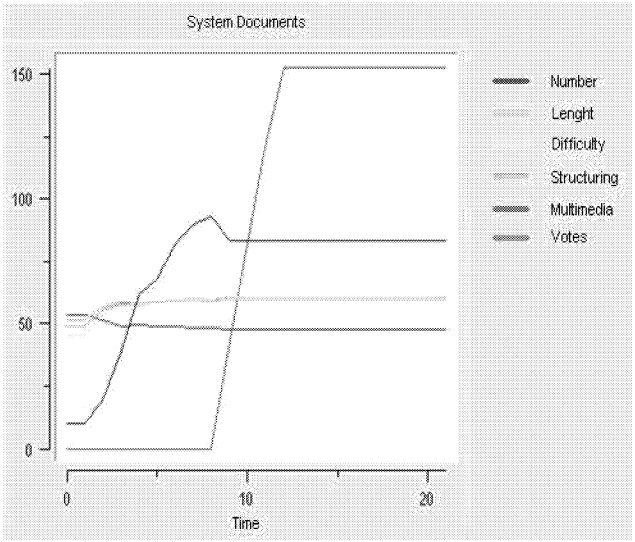


Figure 4: Characterization of documents in the system (1st experiment). Colors represent number of documents, their average length, difficulty, structuring and multimedia, plus the number of votes.

With a lower initial motivation, the evolution of this attribute is slower, affecting the knowledge acquired by the students. With a lower initial frustration resistance, the levels of motivation and knowledge acquired are also affected, but a little less. This may indicate that, even when the student experiences a high level of frustration using the system, if the initial motivation is high, the final results are only slightly affected.

An interesting fact is observed when the initial documents assigned to the students have a high degree of difficulty (third sets of experiments). In the creation phase, the initial motivation does not increase. This is because the majority of the students are not able to understand the reference documents, and thus the documents generated by the students have poor quality. However, in the following phases of the simulation, the motivation and knowledge increase. We assume that this happens because students in a small group, with attributes above the average, are able to understand the initial documents despite their difficulty. This group generates high quality documents, less difficult to understand by the rest of the students. In this way, the average motivation and knowledge increase. From this result we can remark that the cooperation between the agents in the system is positive, because the actions of the agents with better attributes influence the other agents, making them learn more. For the community of learners, this collaboration scheme is beneficial, as the results are better compared with the students working in isolation: they can take advantage of the findings of other students. Thus, the way in which the system works forces an altruistic behaviour

of the participants, in the sense that the documents produced by each student are shared by the others.

5. RELATED WORK

Part of this model was inspired by the work in [K05], which proposes a model of an educational web-system for the study of indirect social interaction, where students can leave recommendations for good learning materials to other students. Our model is targeted to collaborative educational systems, in particular to *KnowCat*. We also use a much more complex model for the students.

Although agent-based simulation is increasingly used for the study of social phenomena, there are not many works in the literature dealing with simulation of web-based educational systems. For example, in [W07] a model is proposed to study the allocation of peer tutors in a community of learners using web-based systems. [YC02] investigates the performance of benevolent behaviour in a web based learning environment (as when individual learners actively provide help to their peers).

There are other works implementing simulated students for their interaction with a real system. For example, in [V02] a simulated student was used to detect "passive students" (students which do not collaborate) and off-topic conversations in a collaborative educational system called *Habipro*. Starting from the student model, we also intend to generate code to test the real web application.

6. CONCLUSIONS AND FUTURE WORK

A simulation model has been developed to study the impact of several parameters on the behaviour of the students using the *KnowCat* educational system. The parameters define student personalities, their friendship relationships, the quality of documents provided by the teachers and the time and tasks assigned in the course. The tool has been tested by means of simple simulations whose results correspond to the expected behaviour.

However, to perform a better analysis of the tool, we intend to compare the results of the simulation tool with those of the actual *KnowCat* system, when used by real students. This study must still be done, although at this point there is a certain amount of raw data available, from which conclusions can be obtained about student preferences and the appropriate values for the parameters we are considering. Another line for future work is the improvement of the tool to represent educational systems which follow paradigms different from those used by *KnowCat*. Finally, we intend to generate code from the models (in particular from the students model) in order to test the real system with different student behaviours.

7. REFERENCES

[AC99] Alamán, X., Cobos, R. 1999. "*KnowCat: a Web Application for Knowledge Organization.*" Proc. of the Int- Workshop on World-Wide Web and Conceptual Modeling, Lect.Notes in Comp.Sc.1727, pp.348-359.

[GT99] Gilbert, N., Troitzsch, K. 1999. "*Simulation for the Social Scientist*". Open University Press.

[G94] Grudin, J. 1994. "*Computer-supported cooperative work: Its history and participation*". IEEE Computer 27(5): 19-26.

[JSW98] Jennings, N.R., Sycara, K., Wooldridge, M. 1998. "*A Roadmap of Agent Research and Development*". Autonomous Agents and Multi-Agent Systems, 1, 7-38 Kluwer Academic Publishers.

[K05] Koper, R. 2005. *Increasing Learner Retention in a Simulated Learning Network Using Indirect Social Interaction*. Journal of Artificial Societies and Social Simulation (JASSS), Vol. 8:2, Mar. 2005. Available at: http://jasss.soc.surrey.ac.uk/8/2/5.html

[MBLA96] Minar, N., R. Burkhart, C. Langton, and M. Askenazi. 1996. "*The Swarm simulation system: a toolkit for building multi-agent simulations*". Working Paper 96-06-042, Santa Fe Institute, Santa Fe. See also the S07 Swarm home page: http://www.swarm.org.

[V02] Vizcaino, A. 2002. "*Can a Simulated Student Detect Problems in Collaborative Learning Environments?*". Proc. IEEE Int. Conf. on Advanced Learning Technologies (ICALT 2002), pp: 244-249.

[W07] Westera, W. 2007. "*Peer-Allocated Instant Response (PAIR): Computational Allocation of Peer Tutors in Learning Communities*". Journal of Artificial Societies and Social Simulation vol. 10, no. 2, 5, available at: http://jasss.soc.surrey.ac.uk/10/2/5.html

[W99] Wooldridge, M. 1999. "*Intelligent Agents*". In "Multiagent Systems. A modern approach to Distributed Artificial Intelligence" (Weiss ed.). pp. 27-77, MIT Press.

[YC02] Yang, J-T D, Chang, T-Y. 2002. "*BIBLES: Benevolent Behavior* In *A Web Based Learning Environment Simulator*". Proc. IEEE Int. Conf. on Advanced Learning Technologies (ICALT 2002), pp:8-13.

SIMULATION ARCHITECTURES AND APPLICATIONS

Norm Selection Through Simulation in a Resource-Gathering Society

Daniel Villatoro
Jordi Sabater-Mir
Artificial Intelligence Research Institute (IIIA)
Spanish Scientific Research Council (CSIC)
Bellatera, Barcelona, Spain
email: {dvillatoro,jsabater}@iiia.csic.es

KEYWORDS

AI-supported simulation, AI in simulation, Intelligent simulation environments, Model design, Parameter Identification

ABSTRACT

In this work we present a mechanism designed for the selection of a suitable set of social norms that would regulate a simulated virtual society. The approach taken for this selection problem is a combination of Genetic algorithms and Simulation. A resource-gathering society has been built using Repast, and on top of it, the genetic algorithm that finds the social norms.

Introduction

Normative systems control the behaviour of the individuals in a certain society by means of norms. These norms, when enforced by the system, formally have the status of *laws* and is the system itself (by means of authorized representatives) who takes care of their observance and fulfilment. However *laws* are only a subset of the norms that work in a society of intelligent and autonomous entities. The system cannot control the behaviour of the individuals all the time and for all the situations and to fill this gap, *laws* are complemented by *social norms*. As Axelrod noticed Axelrod (1986), social norms and laws are mutually supporting: *"Social norms are often best at preventing numerous small defections where the cost of enforcement is low. Laws, on the other hand, often function best to prevent rare defections because substantial resources are available for enforcement"*.

A *social norm* is a rule that is socially enforced by the own non-institutional individuals that are part of the society. In other words, every individual is a potential norm enforcer for the others. The sanctioning applied to the breaking of *social norms* is guaranteed by some kind of social punishment like ostracism or decrease in the reputation instead of the more direct sanctioning associated to *laws*. The nature of *social norms* allow their effects to arrive where *laws* cannot and both mechanisms are complementary. An example of this in every day life is the *supermarket line*: imagine your common supermarket where you do your groceries every time you need a product. There exists a *law*, imposed by the institution and enforced by the cashiers, that tells us that we have to pay before leaving. However nothing is said by the institution about how the queue has to be formed in the cash desks. How have we (as intelligent agents) solved the absence of rules that should guide our behaviour in that situation? Faith in reciprocity have made us come out with a set of *social norms*: we have to follow a FIFO approach in the queue formation and every time you are carrying a heavy loaded cart, and some one behinds you in the queue just carries a light basket, you will let this person pass forward you, hoping that in the exact contrary situation (you carrying a light basket, and the person forward carrying a heavy loaded cart), you will be allowed to pass forward. People that do not share these *social norms* will not receive the benefits that it reports, and no one will let this person go first, because he did not allowed anyone before. Both sets of rules (*laws* and *social norms*) are complementary in order to make shopping in a supermarket a satisfactory experience for all the participants.

In this work, we use an evolutionary approach to find the set of norms (grounded in the idea of altruism and the notion of *image*) that works well in a scenario where autonomous agents have to survive given a limited number of resources. In order to able the testing of these norms in an scenario, we will make use of simulation techniques to build it.

The document is organized as follows. Section presents the scenario that we have chosen as the metaphor for our proof-of-concept. In section gives a deep explanation of all the computational tools that we will use. Section describes all the experiments that have been carried out with their obtained results. Then, in section we do a summary of the work that have been done extracting some conclusions. Last, but not least, in section we explain the future work planned to follow this research.

Description of the Scenario

In this section we will present the scenario we will use in the experiments. We have used a metaphor of a problem already treated by Paolucci et al., based on the methods a society develops for their survival and perpetuation through the exchange of resources.
In the work of Paolucci et al. Paolucci et al. (2006), the authors study the role of groups in the evolution of the concept of altruism. Using a real example from nature, they make simulations of a model of food sharing in a vampire bat society in order to reproduce the same behaviour observed in the real bats. Paolucci et al. show that the concept of altruism in this scenario is essential for the survival of the species as demonstrated by the study of real colonies of bats, and confirmed by the simulations. We adopt the notion of altruism also in our scenario that is based (with some modifications to fit our purpose) in the bats scenario. We are inspired by the "Altruist Vampire Bat Problem" metaphor to build an scenario where we can study the norm selection process. The scenario where we will test the norm selection process is the following:

"Find and Share" Game

Suppose a given society of agents which feed themselves by collecting resources. The resources are distributed randomly in the environment and are easily detectable by the agents. Once an agent finds a resource, it collects it automatically and in this way gains a certain amount of energy, that lengthens its life for a certain period of time and makes the agent seem fatter and bigger. The agents can detect if any other agent around has eaten or is hungry, because the obvious bigger size of the satisfied agent and the smaller size of the hungry one. The agents also have the ability of transferring some of their own energy to another agent.
If a couple of agents meet in the environment, they interact. When interacting, they have two options: transfer some energy to the other agent, or, they can decide to do nothing and walk away.
We suspect that resource distribution is correlated to the emergence of altruism when a certain parameter, such as the maximizing average life expectancy of agents or minimizing the number of death agents, want to be optimized. In other words, we expect that the set of norms becomes useful in very specific configurations of the scenario.
In our case, we will study the set of norms that extend the average life expectancy of the agents, and that, in the same way, minimizes the number of deaths in the society. To achieve this goal, we expect that the set of norms to be selected, by the norm selecting mechanism, promotes altruism and cooperation.
Summing up what have been explained so far: we will have n agents, located in a bi dimensional grid, where the agents will move every time step to an adjacent cell trying to find resources or another agents with whom interact. Every step an agent makes, it consumes one of its energy units, that are restored when resources are found. In case of an interaction with another agent, it can transfer a certain amount of energy. Therefore, a norm will be a set of combined observables, which will define a possible situation, with an action attached to that situation. The ethic code will be the set of all the norms.

Elements of the Simulation

In order to be able to represent the explained problem we need to use several techniques that have been combined in order to accomplish the desired experiments. In this section we will explain how the Repast model has been built, as well as we explain the structure and operation of the Image System. Next, as one of the most important parts of this research, we explain the Norm Selection process done with a genetic algorithm.

Population and Environment: a Repast Model

One of the basic ingredients for our experiment is the population that will assume and follow the norms, and the environment where they will interact. Repast is an Agent Based Simulation Platform that allows to quickly build complete models of simulation. All the needed tools such as graphical interface, plotting tools, and basic skeleton of a model of simulation, are provided.
The Repast model is based upon three main basic components:

- *Space*: The space has all the characteristics of the "physical" environment where the agents are to act.
- *Model*: The way in which how the world works, and how the simulation will be run are explicitly defined in the model.
- *Agents*: In this component of the Repast Model is where the behaviour and capabilities of the agent are explicitly defined.

Image System

Each of our agents is provided with a memory of "known" agents. Basically, each agent is able to keep track of the result of the interactions with all the other agents present in the simulation. Agents have an image ("own believed evaluation of a target" Conte and Paolucci (2002)) of each one of the other agents. In our particular example of the recollecting agents, we want the agents to know the moments that make the image of another agent increase or decrease: An agent x will increase the image of another agent y if y makes energy

transference to x, enlarging then x life's expectancy. On the other hand, the image that an agent x has of another agent y will decrease in two situations:

1. When x, knowing y, transfers energy to y. This way we simulate a basic credit system of *image points*, and ensure that an agent that receives energy from another will have to "return" that energy back to the other to be even, in "image" terms.

2. When x (being starving) meets y (not being starving) and B_y does not offer an energy transference.

At the beginning, we have considered that being unknown is the same that being known with good image, in other words, at the beginning our agents trust.

The Norm Selection Process: a Genetic Approach

We choose an evolutionary approach to find the ethic code that suites better to the given society. We would like to remark that the process of norm selection is totally dependent to the configuration of the world. It seems obvious that a given set of rational agents will not behave in a small populated environment in the same way that they would do in a larger less populated environment, as well as in an environment rich of resources or in a poor environment. For that reason, we will maintain fix all the parameters of the simulation and then study how the norm selection process behaves when changing one of these parameters. Genetic algorithms seem to be the most suitable technique to find this ethic code.

Representation
One of the most important decisions to make before applying a genetic algorithm to a problem is to decide how a solution is represented. Traditionally, a chromosome is encoded as an array of bits where each bit represents a feature of the solution, a.k.a. gene. This boolean representation fits our case where each gene position represents a situation, which in our formal definition of the problem in section was defined. The genetic algorithm will assign a boolean value, that will correspond to the action to take in that situation (`false` means *do nothing*, `true` means *share food*). As a result of assigning to every different possible situation a certain action, we are defining all the norms that will be used.

Fitness function
Evaluation of chromosomes is as important as their representation. The fitness function evaluates a chromosome assigning it a fitness value that tells how good the solution is represented by the chromosome. In our specific problem, the chromosome will have, as genes, all the possible situations that a couple of agents could find themselves (defined by a combination of the observables described in section). The genetic operators will be in charge to change the action to take in a certain situation.
However we have to deal with a very important problem directly related to the nature of our scenario. As we are simulating, many random parameters are used, so, even the same simulation with the same individuals and parameters, executed twice, would obtain different results, similar but different. A non - deterministic fitness function decreases the effectiveness of the GA. To solve the problem of the non-determinism we have adopted the following solution: every time a chromosome is evaluated, the fitness value associated to it is the average of the results of 10 simulations. This value is stored and in case the same chromosome has to be evaluated we recover the value previously calculated. This approach, from our point of view, is an elegant solution to the problem of the non-determinism of the fitness function of the GA, and also, reduces the number of cases being evaluated, which would require higher computational times. In this way, we have provided to the basic GA the capability to deal with non deterministic fitness functions.

Cycle of evolution
The GA starts with an initial population of randomly generated chromosomes and enters into a cycle of evolution. In each iteration, a new generation is produced from within the old one by using evolutionary methods. The fittest chromosomes have more chances to be selected as parents and/or survive in the next generation. The loop is as follows: We first select a number of chromosomes from the existing population to be passed directly to the next generation (a.k.a. survivors). Then from the same population we select an even number of chromosomes to be used as parents. We apply crossover operation to each pair of parents to produce offspring chromosomes. Depending on the probability rate of mutation, we mutate a fraction of the newly produced offspring. Finally, together with the selected survivors, the offspring form the next generation for the cycle.
The evolution continues till a termination criterion is satisfied (i.e. satisfying a fitness threshold or exceeding a number of maximum generations allowed)
We also incorporate the concept of elitism in our GA by passing the best chromosome in the population to the next generation, thus we prevent losing the best solution found. We have observed that using elitism increases the convergence of our GA.

Genetic Operators

- **Selection:** We use the Fitness-Proportionate Selection (a.k.a. Roulette Wheel) method to select survivors and parents. In this method, the fitness value of the chromosome marks its probability to

be selected.

- **Crossover:** There are a handful of crossover implementations that would work for us. We opted for 1-point crossover since it is the simplest and we have not reached any better convergence with other crossover implementations.
- **Mutation:**The mutation operation helps us to avoid falling into local optima in the search space. Mutation takes place by flipping the value of a gene in an offspring chromosome.

Experimental Setting and Results

In this chapter we present all the experiments that have been executed and analyzed with the platform described in subsection . In order to make easier the task for the reader, we have divided the adjustments of the parameters in three groups, parameters related to: the genetic algorithm, the world configuration, and the simulation.

Setting Up the Genetic Algorithm

The genetic algorithm has been fixed with a specific set of parameters DeJong and Spears (1990) which in the literature have been probed to suit in problems with similar characteristics: **Population size** = 50 ; **Number of generations** = 20. De Jong and Spears recommend to set the number of generations to 1000, which we did at the first tests. After observing that the convergence in the worst case was produced below 20th generation, we decided to fix it to 20. ; **Crossover type** = one point ; **Crossover rate** = 0.6 ; **Mutation rate** = 0.001. We have checked this parameters to work fine for our purpose obtaining satisfying results, that will be shown later.

Setting Up the World

As explained in subsection , we have developed an altered Repast model to combine the simulation power of Repast with the genetic algorithm. We have assumed that all the agents will behave in the same way, in other words, they will all follow the same set of norms.

Setting Up the Simulation

Directly related to the parameters of the world configuration, some other parameters have been defined to describe the simulation. All the experiments have been executed during 10000 steps of simulation, making all the agents follow the same set of rules.
In order to observe the effectiveness of the ethic code provided by the genetic algorithm, we have prefixed three different behaviours that we consider basic behaviours for the agents, and hence being able to compare those with the obtained by the Genetic Algorithm:

1. **Random**: Expected to be the worst performing behaviour, we allow the agents to choose a random action every time they meet another agent.
2. **Egoist**: Even though it can be considered just like a very general rule, this behaviour implies that no matter what the situation is, no agent will transfer energy to another.
3. **Sharing**: One agent will transfer energy to another, in case the second one is weaker than the first and the fact of transferring energy does not put the former on risk to die.

We have programmed the Repast model to return us several measures that will be useful for the analysis of the results from the simulation when any of the previously defined parameters are modified. Some of these measures are *Average Life Expectancy* of the agents, and the number of *Living Agents*.

Results

We set up three different experiments that have been run in order to test our initial hypothesis: a genetic norm selector is suitable to find an appropriate subset of norms in order to optimize a certain parameter, which in our scenario means minimizing the number of death agents in the simulation and therefore make the society perpetuates (in some experiments we might be interested instead in lengthen the average expected life of agents). At the very first experiments, we thought that most of the parameters were not important, and that the genetic norm selector would just perfectly work for all the situations. This assumption was wrong. We realized that in some configurations of the world, any of the behaviours (genetic, sharing, random or egoist) adopted by the agents do not make a difference, as they will all obtain similar results. The main characteristics of these types of environments are the following: rich in resources spread in small amounts, widely distributed all over the environment, and with a low transference rate. The reasons why in these kind of environments we will not be able to extract any interesting conclusion is because as the agents are likely to find resources easily, it is not really important for them to share any energy. Consequently, they will all behave equally bad. On the other hand, in resource lacking environments, in case they decided to share resources, this sharing would mean only a lengthen of their death throes, instead of helping this dying agent transforming him in a healthy one, with more opportunities to find resources that would lengthen its life.

Experiment 1: A Poor Resource Environment
In Figure 1, we have represented the number of surviving agents along the time from an initial population

ID	Action	Situation		
1	Do Nothing	You generous or Unknown	You Starving	Me Starving
2	Do Nothing	You Mean	You Starving	Me Starving
3	Do Nothing	You generous or Unknown	You Plenty	Me Starving
4	Transfer Energy	You Mean	You Plenty	Me Starving
5	Transfer Energy	You generous or Unknown	You Normal	Me Starving
6	Do Nothing	You Mean	You Normal	Me Starving
7	Transfer Energy	You generous or Unknown	You Starving	Me Plenty
8	Transfer Energy	You Mean	You Starving	Me Plenty
9	Do Nothing	You generous or Unknown	You Plenty	Me Plenty
10	Do Nothing	You Mean	You Plenty	Me Plenty
11	Transfer Energy	You generous or Unknown	You Normal	Me Plenty
12	Transfer Energy	You Mean	You Normal	Me Plenty
13	Do Nothing	You generous or Unknown	You Starving	Me Normal
14	Transfer Energy	You Mean	You Starving	Me Normal
15	Do Nothing	You generous or Unknown	You Plenty	Me Normal
16	Do Nothing	You Mean	You Plenty	Me Normal
17	Do Nothing	You generous or Unknown	You Normal	Me Normal
18	Do Nothing	You Mean	You Normal	Me Normal

Table 1: *Genetic Norms for 1st Experiment*

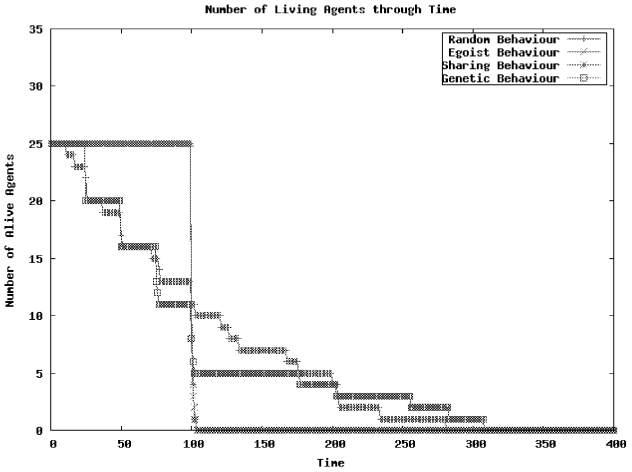


Figure 1: *Surviving Agents in a Poor Environment*

of 25 agents when adopting their different behavioural rules. At the initial state, the agents have 100 time steps of life (in case they do not find any resources which will lengthen their life). The simulation was preset with the following parameters: *Resources Appearance Rate*: 20 step; *Heap Number*: 5 Heaps; *Heap Size*: 1 unit of resources per heap; *Energy Transference Rate*: 25 units of energy. The scenario defined by this parameters represents a resource-lacking world, and also, the fact of an energy transfer do not give an important advantage to the "poor" agent when energy is transferred to them. Once the simulation was ready, we launched the Genetic Mechanism that would find the most appropriate set of norms. These norms can be seen in Table 1.

In this set of norms we do observe some reasonable norms. For example, norms 7, 8, 11, 12, 14 favour the cooperation when another agent is in a worse situation than the first, saving starving agents from death, and making normal agents plenty. Moreover, we observe that this norms are complemented with norms 1, 2, 3, 6, 9, 10, 15, 16, 17, 18 which can be basically resumed in the behaviour that an agent will not transfer energy to another when they are in equal conditions, or the first one is in a worse situation than the second. For the rest of norms, where we have obtained strange results (in the semantic meaning of the norms) we have an easy explanation for it. There is a law inherent to the agent that forgives to it to transfer energy when the agent do not have the amount of energy required for the transference, avoiding the suicides. Therefore, it does not influence the fact that norms 4 or 5 tells the agent to give resources, agents will be told by a superior authority (the environment) that the agents are not able to produce an energy transference (Energy transference rate = 70) when they are starving, which means that their actual level of energy is below 25. As a consequence, in this specific scenario, an starving agent will always have his action imposed (*Do Nothing*) by the environment, without any consideration of the norms. We can observe the results of a comparison of the different behaviours in Figure 1.

We see that, as expected, the Genetic Behaviour (the behaviour that follows the norms established by the genetic algorithm) is the one that extends the society's life for the longest time. The explanation is that the genetic algorithm promotes the cooperation, even though if this cooperation puts in risk the agents life. Observing Fig 1, at the Genetic Behaviour, we see how the population decreases slowly, which means that some "sacrifices" (an agent donate energy even though it will leave itself with a very low level of energy, close to death) are being carried out, for the survival of the specie.
On the other hand, we observe that sharing and egoist behaviours make the society disappear around year (time step) 100, which means that agents were not able to find any resources. Moreover, due to how the sharing behaviour have been designed (sharing is only allowed if after the transference, the "source" agent does not become an starving agent), the sharing agents do have very low probability to share any energy: with an initial life of 100, a transference rate of 70, the starving level set to 25 and very low resources, agents will "not find any resources" and they will only have 5 time steps since the initial instant, to make a successful exchange. Hence we can say, that sharing and egoist in this scenario will behave in the same way. Surprisingly, we observe how the random behavioural agents do a better performance than the sharing or egoist, and the explanation for that is simple: in some situations, the random behaviour makes the agents behave close to the

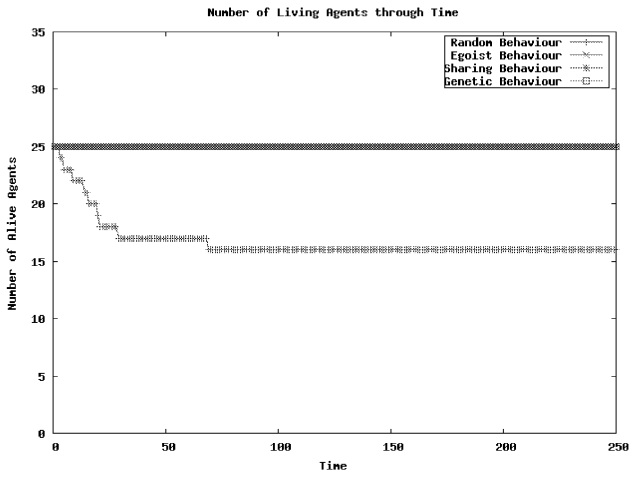


Figure 2: *Surviving Agents in a Rich Distributed Environment*

behaviour they would do in the genetic behaviour.

Experiment 2: A Rich Dispersed Environment

Another very different situation would be if the amount of resources in the environment would allow the specie to perpetuate. Therefore, to achieve this, we set up an experiment where the amount of resources would allow in some situations (depending how agents manage the resources) the specie to survive during the whole simulation. At this point of the research we realized how the distribution of resources would favour the emergence of a set of norms promoting the cooperation. Extending the experiment showed in Fig 1, we increased the amount of resources available to the agents in a simulation with the following parameters: *Resources Appearance Rate*: 1 step; *Heap Number*: 125 Heaps; *Heap Size*: 1 unit of resources per heap; *Energy Transference Rate*: 70 units of energy. This parameters makes our world to be well provided of resources, and the transference rate this time do make a difference when sharing by giving a real opportunity to the dying agent, since after the transference it state will change from *starving* to *normal*. In the same way than in the previous experiment, we launched the Genetic Algorithm to find for us the most appropriate set of norms for this scenario, returning us a set of norms. We do not consider that the analysis of the obtained rules have importance, since for the nature of the scenario, makes the fact of sharing or not sharing resources unimportant, as it can be seen in Figure 2. Nevertheless, the genetic mechanism has found a better performing behavioural norms than the random ones, equalizing the sharing and non sharing performances. We can observe the results in Figure 2.

Our suspicion about the relation of the resource distribution and the sharing importance becomes

ID	Action	Situation		
1	Transfer Energy	You generous or Unknown	You Starving	Me Starving
2	Transfer Energy	You Mean	You Starving	Me Starving
3	Transfer Energy	You generous or Unknown	You Plenty	Me Starving
4	Do Nothing	You Mean	You Plenty	Me Starving
5	Transfer Energy	You generous or Unknown	You Normal	Me Starving
6	Transfer Energy	You Mean	You Normal	Me Starving
7	Transfer Energy	You generous or Unknown	You Starving	Me Plenty
8	Transfer Energy	You Mean	You Starving	Me Plenty
9	Do Nothing	You generous or Unknown	You Plenty	Me Plenty
10	Do Nothing	You Mean	You Plenty	Me Plenty
11	Transfer Energy	You generous or Unknown	You Normal	Me Plenty
12	Transfer Energy	You Mean	You Normal	Me Plenty
13	Transfer Energy	You generous or Unknown	You Starving	Me Normal
14	Do Nothing	You Mean	You Starving	Me Normal
15	Do Nothing	You generous or Unknown	You Plenty	Me Normal
16	Transfer Energy	You Mean	You Plenty	Me Normal
17	Do Nothing	You generous or Unknown	You Normal	Me Normal
18	Do Nothing	You Mean	You Normal	Me Normal

Table 2: *Genetic Norms for 3rd Experiment*

clearer after observing this figure. In such scenario, which is plenty of resources and very well distributed all over the space, agents have a high probability of finding one of these resources that will probably lengthen its life for the same time steps that took the agent to find the resources. Somehow, we have designed an independent-of-norms environment for the agents, where, no matter what they do (share or not), the perpetuation of their specie is ensured. Only in the case of random behaviours we have observed some instabilities and deaths, which we consider reasonable, due to the spontaneous and non-reasonable actions that an agent can decide; for example, a dying agent can meet a plenty agent and decide to transfer the other some energy, giving its last units of energy, remaining therefore in a very close situation to death.

Experiment 3: A Rich Concentrated Environment

Finally, and to confirm this suspicion, we decided to set up an scenario with the following characteristics: *Resources Appearance Rate*: 50 steps; *Heap Number*: 10 Heaps; *Heap Size*: 500 units of resources per heap; *Energy Transference Rate*: 70 units of energy. These parameters are defining a world where resources appears every large intervals of time (almost half the initial life of agents), and, very concentrated although in huge amounts; in other words, it is relatively hard for the agents to find resources, but when they find the resource, they find a huge amount of it. Likewise the two other experiments, the genetic algorithm this time returns us the set of norms in Table 2.

In this set of norms in Table 2 we do observe a promotion of the cooperation. Likewise in the Table

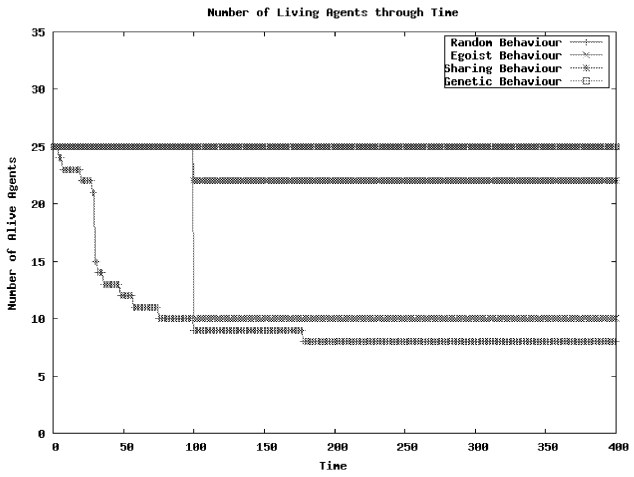


Figure 3: *Surviving Agents in a Rich Concentrated Environment*

1, we can not pay attention to the first 6 rules, that involves acting an starving agent, that, no matter what the norms say, the agent will never transfer energy in that situation, as we have the non-suicide implicit norm. Even so, the rest of norms "enforces" the cooperation. When an agent has a higher energy level than another one, the first one will make a transfer of energy, except in the situation where it can put its life in risk: if the first agent is normal and the second is starving, the first one will only do a transference of energy, if the first agent has a good image of the second. Apart from that, if two agents are in the same energy level, the norm says "do nothing".
The results of the comparison can be observed in the Figure 3

Finally, the suspicion is confirmed. We observe how this time the fact of sharing becomes really important in the society in order them to perpetuate. We observe how sharing and genetic norms make a similar performance, confirming our suspicion that sharing is crucial in some scenarios such as this one. The fact of sharing will make the agents survive for longer, having therefore more possibilities to find resources when those resources are spread again. The population of non sharing and random agents reduces, and the fact that the population reduces, makes that less agents would find food, making the society close to the collapse. It is easily noticeable in Figure 3 that the population in the four cases stabilizes after a time; this is due to after some time steps we suspect that the resources in the world are in equilibrium with the surviving agents, avoiding them to die.
To sum up, the experiments on our platform have allowed us to realize about the restrictions on the norm selection process. The restrictions are inherent to the problem to be simulated, in our case, the relation between the key parameters such as amount of agents, size of the world, food distribution, and food renewal rate. We have analyzed the three scenarios that we consider basic, extracting good results from our norm selection mechanism. Furthermore, we have found a type of scenario where the altruism and the cooperation are promoted by the norms in order to keep the society alive, avoiding the society's collapse.

Conclusions

Norm selection is a useful method in order to find a suitable set of norms that regulates a society in order to optimize a parameter. During the research we have realized that the way of representing the set of norms was a crucial decision. The chosen representation made that our norm selecting process was appropriate to handle with a genetic algorithm.
The integration of all the elements taking part in the norm selection process (Repast model and genetic algorithm) has a complexity we have to deal with. Repast, as a simulation platform, offers many functionalities that are useful during a simulation. However, when Repast is required just to return one value from a certain simulation, launch many different simulations and keep track of the values returned, the simulation platform is not capable. We have had to adapt some details in the structure and implementation of Repast to fully accomplish the task we wanted, having as a result what we could define as a "Genetic Configurable Repast Simulator".
Last but not least, we have extracted some conclusions from the performed experiments. We have been able to see that the cooperation and altruism only make sense in certain configurations of the scenario. These configurations basically are those where the resources are distributed in a very concentrated way with huge amounts and very distant in time intervals. In other scenarios, the altruism is not promoted or it performs in the same way that with an egoist behaviour, although the genetic norm selector still finds the set of norms that behaves better in every different scenario.

Future Work

In this subsection we present all the tasks that will be carried out to continue the work started in here, trying to find answers to questions that still have not been answered with this research, and to answer questions that have arisen during the work.

Brute Force Solution Even though we consider that the Genetic Algorithm is a suitable solution to the norm selection problem, and that we consider the solution given by the GA as a reference for what we will be obtaining with the learning, do we have any reference to compare the results of the GA more than our intuition?

As we do not have any, and do not trust much our intuition in this way, we will focus our efforts in the short term in comparing and analyzing the results obtained by the Brute Force algorithm. The Brute Force algorithm will go through all the possible ethic codes evaluating their functionality and returning the best civil code that optimizes a certain parameter. In this way we could see how far our Genetic Approach was from the global maximum. The Brute Force approach is only taken due to its viability in this simple scenario (as in a bigger one it could not be possible), and to confirm the results with the genetic algorithm.

Different type of agents Another experiment that we want to design is the configuration of an environment were heterogenous agents live, and see how the proportion of normative agents (those that follows the norms) against non normative agents could affect to the variation of the parameter being studied, in our case, the perpetuation of the specie. As we have designed the norms, we give to the agents the opportunity to develop some kind of social exclusion by recognizing their images. Therefore, by introducing heterogenous agents we expect some kind of exclusion emerge for the benefit of one of the agent's groups.

Deep study on the relation of the parameters During this research and the process of experimentation, we have realized that setting up the parameters was a complicated task, due to the direct relation that exists between them when setting up an scenario. For example, it is obvious that the number of agents and the size of the world would determine the density of the population; the density of the population and the amount of food distributed are also related and are key factors for the simulations. Once we find the relation that exists between these parameters, through an exhaustive study of the space of values of those, we will be able to extract better-founded conclusions.

Re-implementation in a different platform The whole process of the norm selection mechanism has been probed to work correctly in a Repast model (Swarm and Java based). In order to be able to affirm the utility of it, we would like to test the same procedure but in different simulation platforms, and if possible, in different programming languages.

Applications

The application domain of this research is directly related to an ongoing research which is carried out by a group of archaeologists. We are presented a non - prehistoric society, already extinguished, known as "the Yámanas". This society was located in Southern Argentina and are one of the groups of the societies commonly known as "canoeros". They lived there for around 6000 years in a very hostile environment. The main success, and reason of study, of this peculiar society is their capacity of auto-organization: the Yámanas were able to auto-organize themselves as a hunter-gatherer society and survive around 6000 years. The archaeologists consider as the hypothesis that the key of the success of this society was due to their strong respect to a known set of social norms. The archaeologists think, that through the use of agent based simulation, they could study the effect of these social norms in two different ways. First of all they are interested in the study of how these social norms were created, which in our research would represent the norm emergence problem. Secondly, they want to study the necessity of all the social norms that once existed, by checking how would the society had worked without one of those norms, and therefore, find what Tennenholtz Fitoussi and Tennenholtz (1998) already called the *"Minimal Social Laws"*, which somehow would correspond to our norm selection mechanism, adding the functionality of selecting it minimal. We find this collaboration work really interesting because it gives us a real domain where to prove our research.

Acknowledgments

This work was supported by the European Community under the FP6 programme (eRep project CIT5-028575 and OpenKnowledge project FP6-027253) and by the project Autonomic Electronic Institutions (TIN2006-15662-C02-01). Jordi Sabater-Mir enjoys a Ramon y Cajal contract from the Spanish Government.

REFERENCES

Axelrod R., 1986. *An Evolutionary Approach to Norms. The American Political Science Review*, 80, no. 4, 1095–1111.

Conte R. and Paolucci M., 2002. *Reputation in artificial societies: Social beliefs for social order*. Kluwer Academic Publishers.

DeJong K.A. and Spears W.M., 1990. *An Analysis of the Interacting Roles of Population Size and Crossover. Proceedings of the International Workshop Parallel Problem Solving from Nature*, 38–47.

Fitoussi D. and Tennenholtz M., 1998. *Minimal social laws*. In *Proceedings of the fifteenth national, tenth conference on Artificial intelligence, Innovative applications of artificial intelligence*. 26 – 31.

Paolucci M.; Conte R.; and Tosto G.D., 2006. *A Model of Social Organization and the Evolution of Food Sharing in Vampire Bats. Adaptive Behavior*, 41, no. 3, 223–239.

A SIMULATION BASED AGENT ARCHITECTURE FOR SUPPORTING DECISION MAKING IN DYNAMIC ENVIRONMENTS

Fabio Palopoli
DEIS, Università della Calabria
and Exeura s.r.l.
f.palopoli@exeura.it

Sergio Greco
DEIS
Università della Calabria
greco@deis.unical.it

KEYWORDS

AI Supported Simulation, Intelligent Agent Architectures, Agent Planning.

ABSTRACT

In this paper we propose an (multi-)agent architecture which allow making complex decisions as well as simple and reactive decisions. As agents operate in a dynamic environment, they use both a rule-based component to react to unexpected events and simulation to make simple decisions. In the proposed architecture the support unit defines plans for agents (complex decisions), whereas agents have a limited power which allows them to make simple decisions such as reacting to events and computing simple plans in a limited space. The main novelty of our approach is that simulation is used in both support unit (to make complex decisions) and agents (to make simple decisions).

INTRODUCTION

In recent years an enormous interest has grown around agent-oriented technology. One of the key problems of the agent building process is defining an effective mechanism for decision making. Clearly, the choice of mechanism for decision making strongly influences the agent architecture modelling process. To introduce our approach to decision making in agents, it is useful to recall an agent definition and, subsequently, an intelligent agent definition. An agent is a computer system that is situated in some environment, that is capable of autonomous action in this environment to meet its design objectives (Wooldridge [2002]). According to the definition above, there is no such thing as an agent disembodied by any environment. An environment can be characterized by some properties, as follows (Russell and Norvig [2003]):

- *accessible* vs. *inaccessible*. An environment is accessible to an agent if the agent can observe every property of the environment state.
- *deterministic* vs. *nondeterministic*. An environment is deterministic if the set $(E \times A)$ only consists of singletons, where E is the set of the possible environment states, and A is the set of actions that an agent should perform in the environment. In other words, an environment is deterministic if and only if every action has a single guaranteed effect.
- *episodic* vs. *non-episodic*. An environment is episodic if "episodes" are independent of one another. In other words, in episodic environments, we can view the agent's experience as divided into independent episodes, which consist of perceiving and acting.
- *static* vs. *dynamic*. An environment is static if it never changes while an agent is deliberating.
- *discrete* vs. *continuous*. An environment is discrete if there are a limited number of distinct, clearly defined percepts and actions.

An intelligent agent is characterized by the following properties:(Wooldridge and Jennings [1995b]):

- Autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state.
- Social ability: agents interact or communicate with other agents
- Reactivity: agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it.
- Pro-activity: agents do not simply act in response to their environment, but, rather, they are able to exhibit goal-directed behavior by taking the initiative. An agent is capable of handling complex, high-level tasks. The decision as to how such a task is best split up into smaller sub-tasks, and in which order and way the sub-tasks are best performed, should be made by the agent itself.

Some of the best known agent concrete architecture are:

- logic based agents (Wooldridge [1997], Wooldridge and Jennings [1995a]) - in which decision making is realized through logical deduction;
- reactive agents (Brooks [1991], Wooldridge and Jennings [1995a]) - in which decision making is implemented in some form of direct mapping from situation to action;
- belief-desire-intention agents (Georgeff et al. [1998], Rao and M. Georgeff [1995]) - in which decision making depends on the manipulation of data structures representing the beliefs, desires, and intentions of the agent; and finally,
- layered architectures (Muller et al. [1995], Muller [1996]) (also called hybrid architectures) - in which decision making is realized via various software layers, each of which more-or-less consists in explicitly reasoning about the environment at different levels of abstraction.

Moreover, the main differences of agent orientation with respect to object orientation is the autonomy of the software components and the possibility to interact with other components. The autonomy allow agents to make decisions about what to do next.

In this paper we propose an (multi-)agent architecture, called $\mathcal{DSA}$ (Decentralized and Simulation based Architecture), which allow to making complex decisions (performed by a support unit) as well as simple and reactive decisions (performed by agents). As agents operate in a dynamic environment, they use simulation in the decision-making process (Kuehne et al. [2005], Lee and Fishwick [1997]). In the proposed architecture the support unit defines plans for agents (complex decisions), whereas agents have a limited power which allows them to make simple decisions such as to react to events and to compute simple plans in a limited space. The main novelty of our approach is that simulation is used in both support unit (to make complex decisions) and agents (to make simple decisions).

Organization of the paper

The paper is organized as follows: in Section we briefly discuss the planning problem; in Section we describe the $\mathcal{DSA}$ architecture and an $\mathcal{DSA}$ agent's behavior; an example is presented in Section . Conclusions and future work are discussed in section .

PLANNING PROBLEM

The scenario we consider is as follows: the goal of the $\mathcal{DSA}$ Agent is to transform the System state $s_0 \in S_0$ (the set of the initial states), into a state $s_g \in S_g$ (the set of the goal or final states). In order to get the state s_g, starting from the state s_0, it executes a plan p, defined by a support unit. Moreover, the agent acts itself as a decision-maker in a real or simulated system and not only as a representative of a decision-maker in a multi-agent simulation. More specifically, the agent reacts to unexpected events and take actions proactively in case that deviations from the plan p occur.

We are given a set of possible states $S = \{s_1, ..., s_n\}$ and a set of actions $A = \{a_1, ..., a_m\}$.

Definition 1 *A plan p is a sequence of actions $a_1.a_2.....a_n$. The set of actions define a transition function and the application of an action a_i to a state s_j, denoted by $a_i(s_j)$ gives a new state s_k. The notation $a_1.a_2(s_j)$ will be used in substitution of $a_2(a_1(s_j))$.* □

In this paper we consider only deterministic actions and plans.

Definition 2 *An agent a is tuple $\mathcal{A} = (S_0, S_g, P)$ where S_0 denotes the set of initial states, S_g the set of final states and P the set of plans, where for each $p \in P$ there is a $s_0 \in S_0$ such that $p(s_0) \in S_g$.*

An active agent $\mathcal{A}g = (s_0, s_g, s, p, q)$ is characterized by an initial state s_0, a final states s_g, the current (or observed) state s, the already executed plan p and the plan to be executed q, such that $p(S_0) = s$ and $q(s) = s_g$. □

An active agent $\mathcal{A}g = (s_0, s_g, s, p, q)$ is in a wrong state if $q(s) \neq s_g$. An influential unexpected event e is an action, not necessarily belonging to A, which is executed before q, i.e. a modification of the plan q into $e.q$. When active agent $Ag = (s_0, s_g, s, p, q)$, evolves into an agent $A'g = (s_0, s_g, s', p, q)$ we say that an influential unexpected event e, modifying the correct state of the agent, has occurred. Analogously, if after the execution of an action a in a state s the new state is $s' \neq a(s)$, we assume that there has been an unexpected event e such that $a.e(s) = s'$.

The problem we consider is the following: given an active agent $Ag = (s_0, s_g, s, p_1. ... .p_j, p_{j+1}. ... .p_n)$, such that $p_1. ... .p_j(s_0) \neq s$, define a plan $q = q_1. ... .q_k.p_i. ... .p_n$, with $n - i \geq 0$, such that $q(s) = s_g$, that is modifying the plan $p_{j+1}. ... .p_n$ into a new plan $q_1. ... .q_k.p_i. ... .p_n$ which allow to get the final state s_g starting from the current state s.

A SIMULATION BASED ARCHITECTURE

In this section we present the $\mathcal{DSA}$ (Decentralized and Simulation based Architecture). A graphical representation of the $\mathcal{DSA}$ is shown in Fig 1. and consists of two main components: *Support Unit* and Agents.

The support unit, which typically is characterized by powerful computational resources, gives assistance to agents, providing plans and up to date data about the environment in which agents are operating.

Agents have only partial knowledge about the environment. The "local" knowledge available to agents is con-

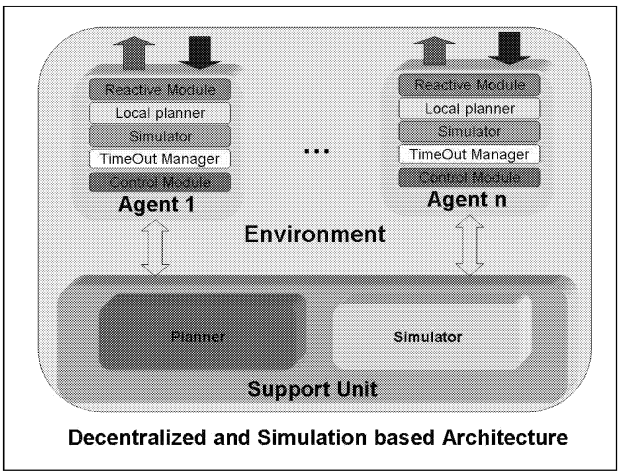


Figure 1. Simulation based Architecture.

tinuously updated using their perceptions and the data received by the support unit.

Moreover, the Support Unit receives agents' observations. Only the agents interacts with the environment. The support unit might give assistance to more than one agent, and agents might request plans to more than one support unit.

Agents are characterized by five main modules:

- A *control module*, whose tasks are: the observation of the environment's state; processing of the observations, action execution, plan monitoring, identification of influential unexpected events, local planning activation; support unit planning activation; information sharing management. Several of these activities are executed in parallel.
- A *local planner*, whose task is to generate a plan. In the next section we will analyze the output of this module.
- A *simulator*, which has to support the choice of the plan.
- A *reactive module*, whose task is to generate a reactive behavior. For reactive behavior we intend the selection of actions by using the current world state as an index into a table of actions, where the indexing function's purpose is to map known situations to appropriate actions. It is triggered by a control module service request.
- A *timeout manager*, whose task is to generate computational time bounds.

We assume that:

- the environment is not completely observable,
- the agent has partial knowledge of the environment before beginning the interaction.

The $\mathcal{DSA}$ Agent behavior is as follows:

At time t_0, in the state s_0, it starts to execute the plan p. At time t_j and state s_j, an unexpected event happens. As a consequence of the influential unexpected event, the Agent $Ag = (s_0, s_g, s, p, q)$ is in a state s, such that $q(s) \neq s_g$. The $\mathcal{DSA}$ Agent strategy in response to this event is as follows:

1. it reacts to the unexpected event so that the agent moves in a ("better") state s',
2. it starts to generate a local plan q' such that $q'(s') = s_g$,
3. it sends to the support unit a replanning request for the current state s'.

Clearly, the computation of the local plan and the request to the support unit are executed in parallel.

Local Planning

The first subtask is to define an ordered set of states, the *state window* $S_w = s_{j-x}, ..., s_j, ..., s_{j+y}$. where $x \leq j$ and $y \leq n - j$. The values of x and y are determined on the basis of the "distance" between the current state s' and the states in S_w, the observation of the environment, and the analysis of the event which has caused the movement of the agent to state s. For instance, if there is an obstacle which does not permit to the agent to reach the state s_j from the state s_{j-1}, the value of x is fixed to 0 so that the window is $s_j, ..., s_{j+y}$. Usually $x \ll i$ and $y \ll n - i$.

Next, the $\mathcal{DSA}$ Agent activates its local planner. The local planner tries to generate a set of ordered plans. In particular, for each $s_w \in S_w$, the local planner tries to generate a plan p_w such that $p_w(s') = s_w$. The plans will be generated in the order defined by the algorithm reported in Fig. 2, where the operators $\succ$ and $\succeq$ are used to compare plans.

In other words, the local planner searches for a plan that would bring the system in a state s_w, which belongs to the set of states S_w. S_w is a subset of the set of states from which, using the prescriptions of the current plan, it is possible to reach the goal state s_g. The selection of the state window and the definition of the order in which the plans are generated are based on the assumption that, given a proximity function, the wrong state s' is typically not far from the state s_i, where $p(s_0) = s_i$. The goal of the local planner is to allow the agent to execute (a part of) the current plan. As we have seen, the local planner will probably generate more than one solution. The solutions will be then evaluated with the support of the simulator. The agent has, in fact, knowledge about the states in the neighborhood of the state s' (using its perceptions and the data received by the support unit) and about the other agents' actual state and behavior. Thus, being the local plan typically very simple, it is reasonable to simulate the behavior of the agents in function of that plan, in order to evaluate in

```
boolean Generate_plans;
begin
    generate plan p_i;
    best_plan = p_i;  new_plan = true;
    h = max(1, x);  k = max(1, y);
    while (h + k < h + y and new_plan and not TimeOut )
do
    begin
        new_plan = false;
        if (k < y) then begin
            generate plan p_{i+k};
            if (p_{i+k} ⪰ best_plan) then begin
                best_plan = p_{i+k};
                new_plan = true;
                k = k + 1;
            end
            else
                k = y;
        end;
        if (h < x) then begin
            generate plan p_{i-h};
            if p_{i-h} ≻ best_plan then begin
                best_plan = p_{i-h};
                new_plan = true;
                h = h - 1;
            end
            else
                h = x;
        end
    end
end.
```

Figure 2. Generation of local plans.

advance the effect of its execution. Using the output of the simulator, the control module chooses the plan p', where $p'.q_w(s') = s_g$, q_w is the portion of the current plan whose preconditions are satisfied by the state s_w, and $p'(s') = s_w$. Thus, a new and effective plan $q' = p'.q_w$ is available and the agent can start to execute it. If it is still waiting for the Support Unit replanning, it communicates to the Support Unit that the previous request must be ignored.

If the local planner is not able to generate any plan, it has to wait for the new plan generated by the Support Unit. In this case, the Support Unit generates a new plan q'' such that $q''(s') = s_g$.

The algorithm reported in Figure 3 shows how the decision module works. In this algorithm, the parallel execution terminates when the execution of one of the two branches is finished plus a time interval used to wait for the execution of the second branch. The function *compare* considers the two possible plans defined by the local planner and by the Support unit. If both are available the best one is selected, otherwise the one available is taken into account.

```
boolean Agent_procedure(executed plan p = [a_1....a_j],
                        plan to be executed q = [a_{j+1}....a_n];
                        initial state s_0; expected state s_j;
                        goal state s_g; final time t_g)
var
    s current state
    t current time
begin
    if (s_0 == s_g) then return true;
    if (s == s_j) then
        if (s_g not reachable from s_j using q in time t_g-t) then be-
gin
            identify a support unit SI
            (q', s_f, t_f) = SI.newplan(s, s_g, t_g)
            Agent_procedure([ ], q', s, s, s_f, t_f)
        end
        else begin
            apply a_{j+1} to the current state s_j;
            let s_{j+1} = a_{j+1}(s_j) the expected state at step j + 1
            Agent_procedure([a_1....a_{j+1}], [a_{j+2}....a_n], s_0, s_{j+1}, s_f, t_f)
        end;
    // an unexpected event e occurred or s ≠ s_i
    if (there is an unexpected e) and (s ≠ s_j) then begin
        react(s, s_j, e);
        parallel begin
            begin
                q' = localplan(s, s_j, p, q, t_g);
                s_f = s_g);  t_f = t_g);
            end
            begin
                identify a support unit SI
                (q'', s_f, t_f) = SI.newplan(s, s_g, t_g)
            end
        parallel end;
        q̂ = compare(q', q'');
        Agent_procedure([ ], q̂, s, s, s_f, t_f)
    end
end.
```

Figure 3. The control module cycle algorithm.

RUNNING EXAMPLE

A mobile robot's goal is to navigate from point A to point B in a two-dimensional space. An incomplete map of the environment and a plan p_0 to travel from a point A to a point B are available (here, the robot being located at points A and B denote, respectively, the initial and goal states). The mobile robot is supported by a Support Unit, which is able to compute faster than the robot, that is counter-wisely characterized by rather restricted computational resources. The mobile robot includes a control unit (with sensors and effectors), a reactive module, a local planner and a simulator. The support unit includes a replanning module. It is not necessary that the robot unit and the support unit are parts of a single physical module. The robot has a limited knowledge about the environment in which it acts, due to the restricted resources available to store data. The robot receives dynamically the data about

the area in which it is located by the Support Unit, and by its perceptions. Indeed, the Support Unit gathers information about the environment sent by the robots which act in it.

At some time t_i, the mobile robot's warning unit detects that an unexpected obstacle lies on its way to B. Assume that it categorizes the obstacle as an influential unexpected event. At this point, the robot stops the execution of the plan p_0 and starts the execution of the reactive behavior (we recall that the selection of a reactive action can be considered an almost instantaneous process). For simplicity, we suppose that the only reactive rule for obstacle avoidance might be as follows:

RULE1:
if (there is an obstacle on your trajectory) **then**
turn 90 degrees to your left;
go in this direction for 20 meters;
end if

Moreover, for simplicity, we assume that the plan p_0 is a sequence of actions $a_0, ..., a_n$, whose preconditions and postconditions are simply as follows:

$precondition(a_i)$=$robot\ in\ (x_{i-1}, y_{i-1})$
$postcondition(a_i)$=$precondition(a_{i+1})$=$robot\ in\ (x_i,\ y_i)$

After the execution of action a_i, due to robot reaction, its location becomes (x'_i, y'_i). This state does not satisfy $postcondition(a_i)$. Let $q = a_{i+1}...a_n$ be the portion of plan that the robot should still execute. $s'_i = Robot\ in(x'_i, y'_i)$ does not satisfy the precondition of a_{i+1}, that is the first action of q. Thus, as $q(s'_i) \neq s_g$, the robot is in a wrong state. In other words, if the robot executes the actions belonging to plan q, it will not reach the point B.

Next, the mobile robot's control unit defines a timeout for the replanning activities and starts two concurrent processes:

- It activates its local planner,
- It sends a replanning request to the support control unit.

The request contains relevant data about the system state at the current time t_i. These data include a description of the obstacle and its location, the location of the robot, its speed, and so on.

The first task executed by the local planner is to define a State Window S_w. In this example, the robot simply identifies an ordered set of points on the "route" defined by the plan p_0, in order to resume its execution. For instance, assuming $x = 2$ and $y = 2$, the robot identifies the set of states $S_w = \{s_{i-2}, s_{i-1}, s_i, s_{i+1}, s_{i+2}\}$.

We recall that s_{i+k} (with $-x \leq k \leq y$) is the set of states that satisfies the preconditions of the action a_{i+k+1}. In this example, for simplicity, we assume that $s_{i+k} = Robot\ in\ (x_{i+k}, y_{i+k})$. Now, in order to define one or more ways to roll back to the original plan p_0, the local planner generates a sequence of plans. Unfortunately, being the environment dynamic, this processes can not continue indefinitely, so it is necessary to comply with the timeout. Thus, until the timeout is not expired, the local planner generates plans in the order discussed in Section 3. In particular, it tries to generate firstly a plan whose initial state is $s'_i = Robot\ in(x'_i, y'_i)$ and the goal state is s_i; we call it $plan_i$. Next, it tries to generate a plan $plan_{i+1}$, whose goal state is s_{i+1}. If $plan_{i+1}$ is "better" than $plan_i$ it computes $plan_{i+2}$ otherwise it computes $plan_{i-1}$, and so on.

In order to evaluate and compare plans and to check their feasibility, it is reasonable to simulate what shall happen in the local area of the environment (that is typically very limited) in which the robot acts (to reach a state belonging to S_w). The Robot, in fact, has knowledge about the local area in which it acts, and about the behavior of the other robots. The aim of the simulation is to evaluate the effective feasibility of the execution of the plan, and, if it is the case, its quality. For example, assuming that $plan_i$ prescribes that our Robot R has to travel on a certain road in the time interval $t_a : t_b$, and that the plan of another robot $R1$ prescribes that it has to work in that road in a way that does not allow any other vehicle to travel through it in the time interval $t_c : t_d$, where $t_c \leq t_a$, and $t_b \leq t_d$. If the plan of robot $R1$ is feasible, then $plan_i$ is not applicable. Moreover, if the plan of robot $R1$ is not feasible, due, for instance, to the actions of other robots which might delay the robot $R1$, then $plan_i$ is feasible and can be executed by robot R.

Thus, the simulator support the evaluation of the plans generated by the local planner. At this point, if at least one plan is generated by the local planner and at least one plan is feasible, the control module chooses the best one, for example, $plan_{i-1}$. For instance, the $plan_{i-1}$, to reach state s_{i-1} from state s'_i, is composed by the sequence of actions $b_1 . . . b_h$, the new plan q', allowing the robot to reach the state goal s_g from the current state s', is composed by the sequence of actions $b_1 . . . b_h . a_i . . . a_n$.

When a new effective local plan is available, it is not necessary to wait for the Support Unit replanning. Thus, after a timeout, it executes the new plan and ignores plans which might have been received from the Support Unit. Moreover, when the decision of applying the local plan has been taken, the robot sends a new message to the Support Unit, in which it communicates to stop the replanning process. Executing the new plan q', if no other unexpected event happens, the Robot will travel from point (x'_i, y'_i) to point $(x_{i-1}, y_{i-1}$ first and, then, to point B. If, during the local replanning, a new plan is provided by the Support Unit, and at least one plan is generated by the local planner and is accepted by the control unit before the expiration of the timeout, the control unit chooses the best one.

Until now, we have supposed that the local planner is able to generate at least one plan that might bring the Robot from point (x'_i, y'_i) to a point (x_w, y_w), such that state $s_w = Robot\ in(x_w, y_w)$ belongs to the state window S_w already defined. Unfortunately, it is possible that

the local planner is not able to generate any of the plans characterized by the properties described above. In such a case the Robot has to wait for a completely new plan generated solely by the Support Unit that, without any other contribution, will bring the Robot from the point (x'_i, y'_i) to the point B.

CONCLUSIONS AND FUTURE WORK

The approach proposed in this paper is based on a new agent architecture, called $\mathcal{DSA}$ (Decentralized and Simulation based Architecture). Simulation seems to be a powerful tool to improve agent decision making processes. We have, essentially, designed an architectural framework. Due to its general and flexible form, we argue that the $\mathcal{DSA}$ approach can be suitably contextualized to several application domains. In this paper, in particular, we have proposed an application of the $\mathcal{DSA}$ approach to the replanning problem. As for future research, more studies are required to optimize the underlying algorithms. A prototype of the $\mathcal{DSA}$ system is currently under implementation at the University of Calabria.

References

R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

M. P. Georgeff, B. Pell, M. E. Pollack, M. Tambe, and M. J. Wooldridge. The belief-desire-intention model of agency. *5th Int. Ws. on Intelligent Agents, Agent Theories, Architectures, and Languages*, pages 1–10, 1998.

R. Kuehne, C. Wille, and R. Dumke. Software agents using simulation for decision-making. *ACM SIGSOFT Software Engineering Notes*, 30(1):1–6, 2005.

J. J. Lee and P. A. Fishwick. Simulation-based planning for multi-agent environments. *Proceedings of the 1997 Winter Simulation Conference*, pages 405–412, 1997.

J. P. Muller. The design of intelligent agents, a layered approach. *Springer*, 1996.

J. P. Muller, M. Pischel, and M. Thiel. Modelling reactive behaviour in vertically layered agent architectures. *Intelligent Agents: Theories, Architectures and Languages, LNAI, Springer*, 890:261–276, 1995.

A. S. Rao and M. M. Georgeff. Bdi agents: from theory to practice. *Proc. Int. Conf on Multiagents Systems (ICMAS-95)*, pages 312–319, 1995.

S. J. Russell and P. Norvig. Artificial intelligence: A modern approach. *Pearson Education*, 2003.

M. J. Wooldridge. An introduction to multiagent systems. *John Wiley and Sons Ltd*, 2002.

M. J. Wooldridge. Agent-based software engineering. *IEEE Proceedings on Software Engineering*, 144(1):26–37, 1997.

M. J. Wooldridge and N. R. Jennings. Agent theories, architectures and languages. *ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam*, pages 408–431, 1995a.

M. J. Wooldridge and N.R. Jennings. Intelligent agents: theory and practice. *Knowledge Engineering Review*, 10, 1995b.

BIOGRAPHIES

SERGIO GRECO received its laurea degree from University of Calabria. Next he was researcher at CRAI, a computer science research consortium. Next he was assistant professor at the University of Calabria. During this time he was visiting researcher at the research center of Microelectronics and Computer Center (MCC) of Austin (Texas) and at the Computer Science Department of University of California at Los Angeles. Currently, he is a full professor at the faculty of Engineering at the University of Calabria and chairs the Department of Electronics, Computer and System Sciences (D.E.I.S.).

FABIO PALOPOLI received its laurea degree from University of Calabria. Currently he is working toward his PhD in Computer Science and is a member of the R&S division at Exeura s.r.l., a spin-off company of University of Calabria concerned with Knowledge Management and general Information and Communication Technologies.

WEB BASED SIMULATION

A Performance Test Platform

R. Heinzl, G. Mach, P. Schwaha, and S. Selberherr

Institute for Microelectronics, TU Wien
Gußhausstraße 27-29/E360, A-1040 Vienna, Austria
E-mail: {heinzl|mach|schwaha|selberherr}@iue.tuwien.ac.at

KEYWORDS
Performance test, benchmark, graphical user interface, programming languages, web-application, Linux, posix, Bash

ABSTRACT

We implemented a fully automated benchmarking system, the performance test platform (PTP). It is developed with special emphasis on flexibility and therefore uses library centric application design combining several orthogonal modules. The modules consist of a base of Bash shell scripts and an orthogonal web front-end written in `PHP`. We thereby ensure that no restrictions are imposed on the benchmarked systems. Furthermore we show the platform's application and results obtained by benchmarking different algebraic equation specification approaches.

INTRODUCTION AND MOTIVATION

We present the performance test platform (PTP), a fully automated benchmark system to consequently analyze compile-time and run-time performance of different algorithms, programming paradigms, or even complete applications. The compiler attributes are fully parametrized, which allows the use of different compilers and options on multiple hosts with several operating systems. It is not only possible to pass various options to the compiler, but it is also possible to pass different options to the newly compiled executable. The rapid development of micro processors and the huge range of currently available processor types requires very complex tools to analyze applications in order to optimize their run-time performance. Additionally, compilers also undergo their own evolution and different compilers introduce a new set of optimization possibilities to control the new features of modern micro processors with each new generation. The possibilities of combining all of these switches and options is increasing even more rapidly and results in a staggering amount of possibilities to choose from. A manual selection of settings to obtain optimal performance is therefore not an efficient task as even different types of applications require a different set of options and switches.

A project was therefore initiated with the goal to determine which compiler-options result in optimal code on different architectures, operating systems, and from different programming paradigms. During the development of a batch-job-system using Bash shell scripts [12] the idea to extend this to a complete performance test platform was born. Once a system consisting of several shell scripts and a hierarchical directory structure was developed, usability had to be increased to make this system accessible on various platforms without manual interaction. A database system, a graphical user interface, as well as graphical statistic modules have therefore been developed.

THE DEVELOPED PLATFORM

The base of the developed system consists of a collection of Bash shell scripts and an orthogonally implemented graphical web front-end in the dynamic web programming language `PHP` [5], cf. Figure 1. Both are supplemented by a `MySQL`-database [13] and `GNU R` [7], a statistical and graphical suite used to process the results and create graphical output.

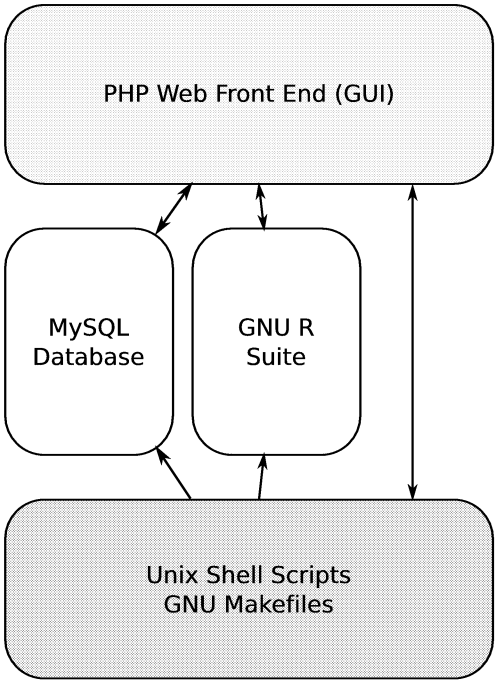


Figure 1: Overview of the performance test platform.

We chose a shell-script backend to ensure the possibility to easily change parameters, templates and scripts manually or by the graphical user interface. We thereby achieve a maximum of freedom and openness for a large variety of systems, languages, compilers, and algorithms to test. The database was implemented in `MySQL`, because it is widely available, easy to use, and very stable when using InnoDB-engines [15, 11] with foreign constraints. Using these features and a

graphical front-end such as `phpMyAdmin` have the possibility to apply even complex changes (i.e. changing the constraints or dependencies of tables) in an efficient and safe way.

We also implemented a graphical user interface module for `GNU R` to finally obtain HTML pages. It provides the facilities to produce graphical output in the style of common spreadsheet-applications from performance test results. The templates used guarantee an easy way to modify the style of the resulting figures.

We chose a web front-end written in `PHP` to control the test platform for many reasons. First, we can use nearly every system providing a browser to control the platform, second we did not need to implement a separate GUI, and, finally, the `PHP` language provides facilities to execute programs on different systems. We can therefore use the web-front-end with other underlying systems as well.

The most important feature of the base system is its modularity and the resulting exchangeability of modules. The main modules can be used without the graphical front-end. This is ensured by using a sophisticated directory structure and small task specific shell scripts allowing modification or substitution of single tasks in the benchmark system or single configurations for one benchmark. The appearance and style of the final figures can thereby be changed without any modification of the core system itself.

The directory structure consists of a directory for each test. This root directory then contains directories for configuration files, source files, and templates, as well as directories for additional files, e.g. input files, concerning the simulation process and for each configuration, option, and host. Figure 2 shows the directory tree of the test-directory.

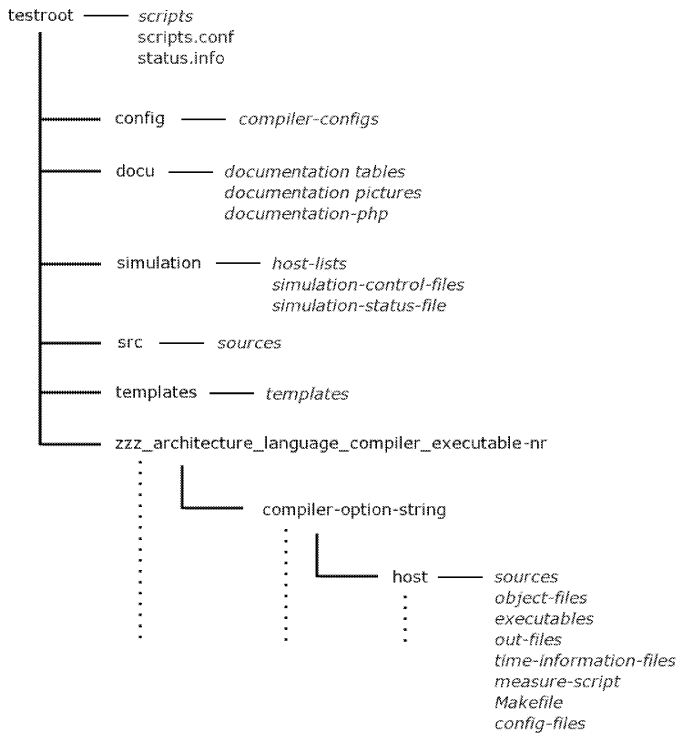


Figure 2: The directory tree of the PTP.

The `config` directory contains all configuration files for a job, where each test (different language, executable, or compiler) has its own configuration. The following table gives an overview of some of the possible tags.

Tag	Explanation
xxxARCHITECxxx	architecture of the system
xxxLANGUAGExxx	language of the tested program
xxxHOSTxxx	space-separated list of hosts to use for the tests

The `template` directory contains templates for Makefiles and time-measure scripts. The time-measure scripts are invoked by the `runonhost.sh` script and measure the compile time as well as the execution time. The `src` directory contains the sources for all tests of this job (different programs, compilers, and hosts). The `zzz_*` directories are named according to the used architecture, the language, the compiler, and a unique executable number. These directories contain one subdirectory per compiler-option-string. Each of these directories contains one directory per host. There are several files in these directories: `makefile.conf` and `measure.conf` are the configuration files for the `Makefile` and `measure.sh` script. Additionally, the source files along with the resulting object files and executables and, after the tests have been run, the files containing the test results are located here.
The `simulation` directory contains, beside a few control files such as `simulation.started` and `simulation.ended`, the status file `simulation.status` and for each host the job lists and output files. The simulation can be controlled by editing the host job lists or the two control files.
After running the tests from the web-front-end, the `docu` directory contains all tables, figures and a `PHP`-file, which is included in the front-end-documentation. In this way it is guaranteed, that the documentation is accessible from the front-end in the typical style, but can also be used without the front-end.
The shell-scripts mirror the procedure of the benchmark test:

- First the directory structure is created and set up.
- The files are then distributed and the actual tests are run.
- The final task is to document the results.

THE WEB-FRONT-END

`PHP`-Scripts control the basic layout, the login (see Figure 3), the connection to the database as well as the report section, the control section, and the tutorial section.
Authentication is performed against the login management of the server (i.e. local accounts using `PAM` [1] or network accounts using `NIS` [2]). Therefore no separate login-management nor accounts have to be pro-

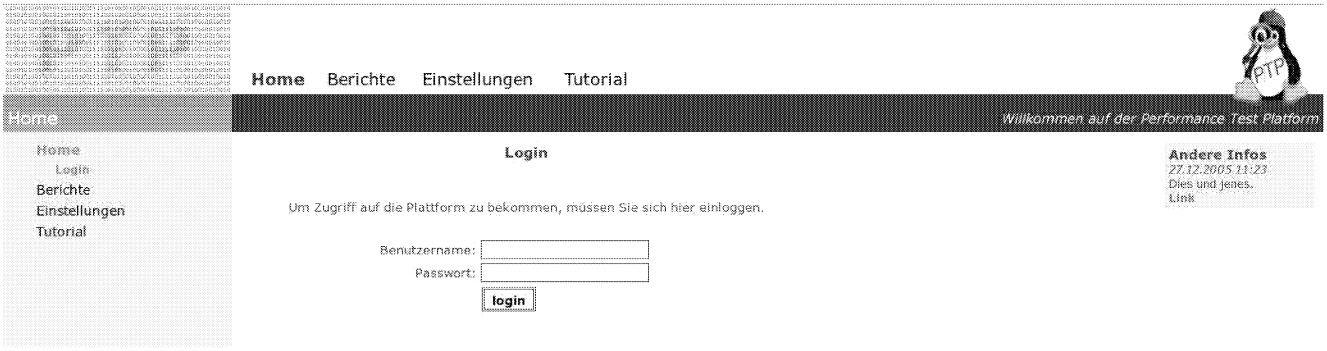


Figure 3: Login to the performance test platform.

vided, but all users having valid logins at the server can login to the platform out of the box. The biggest advantage of this login management is its modularity, so even advanced authentication systems such as `kerberos` [3] can easily be incorporated. Cookies are used to avoid outdated login sessions. The basic graphical layout of the PTP is different for normal users and for admins, respectively.
All reports concerning hosts, jobs and the platform can be found in the report-section ('Berichte'). Access to reports dealing with the platform is restricted to admins only, all other reports are viewable by every user of the platform.

Name	OS	CPU	Memory	Select
in3	linux	PIV 2400MHz	1GB	
in30	linux	PIII (Coppermine) 1000MHz	512MB	☑
in31	linux	PIV 2400MHz	1GB	
in32	linux	Celeron (Coppermine) 800MHz	256MB	
in33	linux	Athlon 64 3500+ 2200MHz	2GB	
in34	linux	PIV 2400MHz	1GB	
in36	linux	PIV 3400MHz	3GB	
in37	linux	PIV 3200MHz	2GB	
in38	linux	PIV 2400MHz	1GB	
in39	linux	PIV 2400MHz	1GB	

Verfügbarkeit der ausgewählten Hosts prüfen: OK

Figure 4: List of hosts in report section.

Reports about the hosts (Figure 4) contain the names of the hosts, their operating systems, the CPU type and speed, and the amount of main memory. Information on the availability of selected hosts which can be used by the system for benchmarks is provided on a separate page (see Figure 5).
Reports about the jobs contain the root-directory of the job and its current status. Information about logins is available in the platform report section.
All controls about hosts, jobs, and the platform can be found in the control section ('Einstellungen'). A user can change only the settings of his own jobs.

Name	Ping	SSH	Load
a09	Host unknown	n/a	n/a
in30	Host unreachable	n/a	n/a
in32	OK	Error	n/a
in36	OK	OK	5:46pm up 86 days 21:35, 42 users, load average: 0.00, 0.00, 0.00

Figure 5: Overview of the availability of selected hosts.

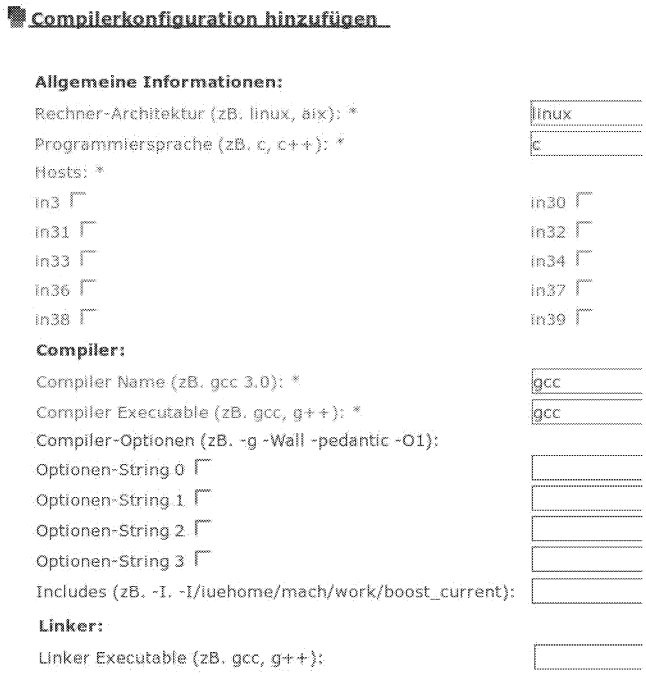


Figure 6: Mask to add a configuration: the red parts are mandatory.

The mask for adding configurations (Figure 6) refers to the configuration tags shown earlier in this paper. It is divided into several sections:

- The section 'Allgemeine Informationen' deals with the architecture and the language and the selection of hosts that should be used.

- The 'Compiler' section contains the name and executable of the compiler. Compiler option strings and include paths can be specified optionally.
- The next section is called 'Linker'. If this entry is empty, the compiler executable is assumed to be used for linking as well. There are optional fields for linker flags, library paths, and libraries to link against.
- In the section 'Source-Files' the names of the source files along with the header files needed to compile the program are specified.
- In the last section 'Make and Executable' it is possible to specify the name of the make command and the executable. Flags and switches for the make command or the executable option strings can also be entered here.

The important part of analyzing the results of the performance test runs is managed by a wrapper tool to `GNU R`. Different templates are used to interpret the corresponding output of the single tasks obtained by the shell scripts. Transformations of the output data, selections, and finally illustration templates have been developed to ease the complex process of intelligence collection. The 'Analyze Results' mask is shown in Figure 7.

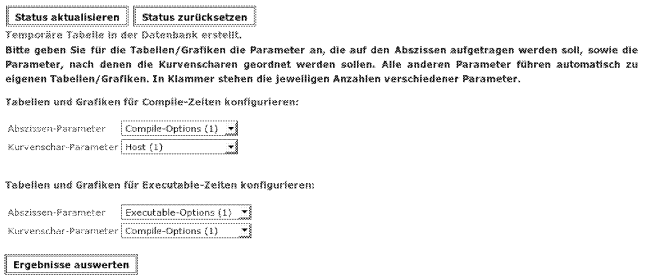


Figure 7: The 'Analyze Results' mask.

The parameter for the abscissa and the set of curves can be chosen for compile times as well as execute times. All other parameters result in separate tables and figures based on given and adjustable templates.

RESULTS OF A BENCHMARK

Using the performance test platform we compared different equation specification approaches, based on a generic scientific simulation environment [9, 14, 10] and the corresponding performance analysis [8]. Therefore we analyzed several techniques which are available in C++.

CPU type	Clock speed	RAM	Compiler	MFLOPS
P4	2.8 GHz	2 GB	GCC 4.0.2	2310.9
Core Duo	2.4 GHz	2 GB	GCC 4.3	8804.0
Athlon 64	2.2 GHz	2 GB	GCC 4.1.2	10134.0

We compared the following equation specification methods:

- The expression template technique was used as described in [4].
- The GNU g++ implementation of the `valarray` data type was used as a test.
- Blitz++ [16]
- Naive C++ implementation
- The GSSE approach [9] based on topological traversal and an equation concept based mostly on the Boost Phoenix library [6].

The test is performed using a vector addition $A_f = A_b + A_c + A_d$, evaluated with different vector sizes. The definitions of the corresponding data types are given in the next code snippet:

```
typedef std::valarray<double>   Array_t;
typedef blitz::Array<double,1>  Array_t;
typedef std::vector<double>     Array_t;
```

The next figures presents and compare the results obtained by the different approaches. The y-axis is labeled with operations per second. The vector addition is built by three operations, two additions and one assignment. Figure 8 shows the high runtime performance of an Intel P4 microprocessor as long as the values are cached.

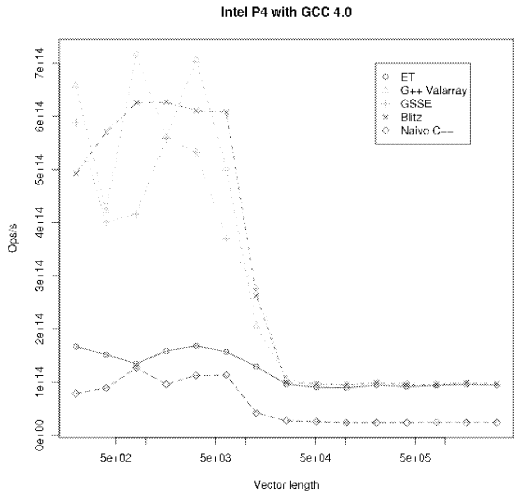


Figure 8: Comparison of functional specification on an Intel P4 microprocessor.

Figure 9 presents the results for an Intel Core Duo microprocessor. The sudden drop in runtime performance at certain vector lengths should be noted.

Figure 10 depicts the best benchmark results for an AMD Athlon 64 microprocessor with AMD rating 3500+.

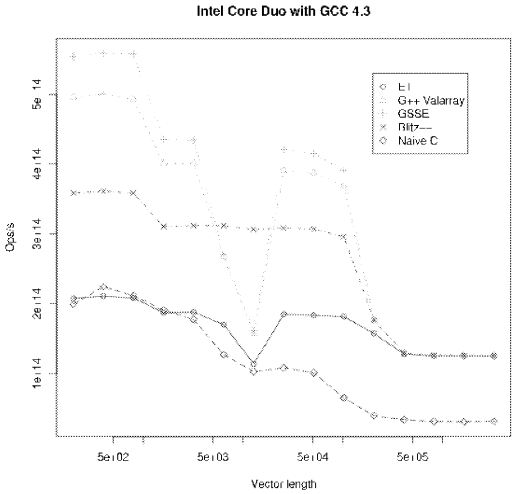


Figure 9: Comparison of functional specification on an Intel Core Duo microprocessor.

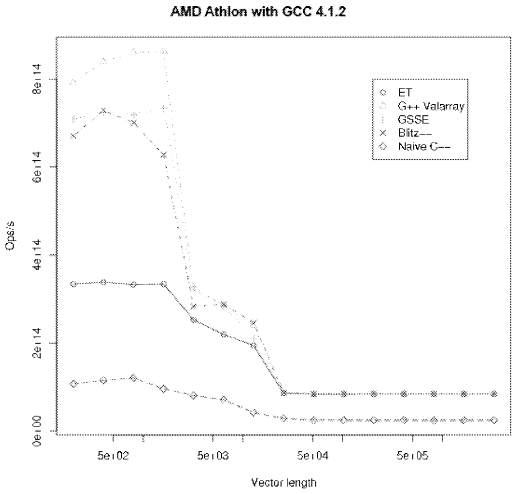


Figure 10: Comparison of functional specification on an AMD Athlon 64 microprocessor.

CONCLUSION

Based on a set of Bash shell scripts, a statistical analysis tool, and a comprehensive graphical user interface, a performance test platform was developed to ease the benchmarking processes. A completely automated process chain is thereby obtained to analyze the performance optimization possibilities of modern microprocessor architectures and new compiler versions.

REFERENCES

[1] *Pluggable Authentication Module (PAM)*, 1995. `http://www.kernel.org/pub/linux/libs/pam/`.

[2] *Network Information System (NIS)*, 2004. RFC 3898.

[3] *Kerberos: The Network Authentication Protocol*, 2007. `http://web.mit.edu/kerberos/www/`.

[4] D. Abrahams and A. Gurtovoy. *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond (C++ in Depth Series)*. Addison-Wesley Professional.

[5] Achour, Betz, Dovgal, Lopes, Olson, Richter, Seguy, Vrana, et al. PHP Manual. http://www.php.net/manual/en/, 2006.

[6] Boost Phoenix 2. *Boost Phoenix 2*. Boost, 2006. http://spirit.sourceforge.net/.

[7] R Foundation. The R Manuals. http://cran.r-project.org/manuals.html, 2007.

[8] R. Heinzl, P. Schwaha, M. Spevak, and T. Grasser. Performance Aspects of a DSEL for Scientific Computing with C++. In *Proc. of the POOSC Conf.*, pages 37–41, Nantes, France, July 2006.

[9] R. Heinzl, M. Spevak, P. Schwaha, and T. Grasser. A High Performance Generic Scientific Simulation Environment. In *Proc. of the PARA Conf.*, page 61, Umea, Sweden, June 2006.

[10] R. Heinzl, M. Spevak, P. Schwaha, and S. Serlberherr. Performance Analysis for High-Precision Interconnect Simulation. In *Proc. of the ESMC. Conf.*, pages 113–116, Toulouse, France, 23-25. October 2006.

[11] A. Heuer and G. Saake. *Datenbanken: Konzepte und Sprachen*. mitp, 2nd edition, 2000.

[12] M. Cooper. Advaced Bash Scripting Guide. http://www.tldp.org/LDP/abs/html/, 2006.

[13] MySQL AB. MySQL 5.1 Reference Manual. http://dev.mysql.com/doc/refman/5.1/en/index.html, 2006.

[14] P. Schwaha, R. Heinzl, M. Spevak, and T. Grasser. Advanced Equation Processing for TCAD. In *Proc. of the PARA Conf.*, page 64, Umea, Sweden, June 2006.

[15] H. Tuuri et al. The InnoDB Documentation. http://www.innodb.com/support/documentation, 2007.

[16] T. L. Veldhuizen. *Active Libraries: Rethinking the Roles of Compilers and Libraries, in* Proc.ISCOPE'98, Lecture Notes in Computer Science. Springer-Verlag, 1998.

BIOGRAPHIES

RENÉ HEINZL studied electrical engineering at the Technische Universität Wien. He joined the Institute for Microelectronics in November 2003, where he finished his doctoral degree in September 2007. In April 2005 he achieved first place at the doctoral competition at the EEICT in Brno. His research interests include computational science, programming paradigms, high performance programming techniques, process simulation, solid modeling, scientific visualization, and algebraic topology for TCAD.
GEORG MACH was born in Vienna, Austria, in 1979. He studies electrical engineering at the Technische Universität Wien. He joined the Institut für Mikroelektronik in June 2003, where he is currently working on his diploma thesis.
PHILIPP SCHWAHA studied electrical engineering at the Technische Universität Wien. He joined the Institute for Microelectronics in June 2004, where he is currently working on his doctoral degree. His research activities include circuit and device simulation, device modeling, and software development.
SIEGFRIED SELBERHERR received the doctoral degree in technical sciences from the Technische Universität Wien in 1981. Since that time he has been with the Technische Universität Wien as professor. Dr. Selberherr has been holding the "venia docendi" on "Computer-Aided Design" since 1984. As of 1988 he has been chair professor of the Institut für Mikroelektronik. From 1998 to 2005 he served as Dean of the "Fakultät für Elektrotechnik und Informationstechnik" at the Technische Universität Wien. His current topics of interest are modeling and simulation of problems for microelectronics engineering.

Labtool - A Managing Software for Computer Courses

R. Heinzl, G. Mach, P. Schwaha, and S. Selberherr

Institute for Microelectronics, TU Wien
Gußhausstraße 27-29/E360, A-1040 Vienna, Austria
E-mail: {heinzl|mach|schwaha|selberherr}@iue.tuwien.ac.at

KEYWORDS

computer courses, managing software, monitoring and grading system, web front end, `LDAP`, `kerberos`, `PHP`

ABSTRACT

A comprehensive tool to manage computer courses was developed relying on a library centric application design. Backward compatibility to existing systems, and special emphasis on usability were also part of the design process. The experience gathered from implementing monitoring and controlling modules for the heterogeneous hardware and software system was applied to make the system error tolerant. To this end we specified, developed, tested, and evaluated several modules and a graphical user interface and finally integrated them into a programming computer course at our university.

INTRODUCTION AND MOTIVATION

The basic question we faced at the beginning was, how to automatically organize, e.g. a programming language course for 200 to 300 students in groups to 30? More importantly, how can a heterogeneous system with various services and tasks be implemented in a stable fashion that does not interfere with a continuing evaluation of students? We therefore summarize the most important issues in developing software for computer courses from the view of a programming language lecture for the first and second semester at a university:

- availability of the system needs to be at 100 percent during the courses
- automation of various tasks such as tracking the progress of students, report generation, and system service overview
- limited resources such as instructors, and computers
- appointment constraints like other courses (on both sides - students and teachers)
- necessary data exchange with other departments of the university or other programs to administrate the course

Since we did not find any free graphical tools to avoid excessive paperwork and security holes in a modular, redundant, backward compatible fashion with the administrative databases, we decided to develop and implement our Labtool. Not only the finite resources but also social aspects had influence on the specific design. We present our implementation, focusing on the encountered problems and their solutions.

A GENERALIZED APPROACH

One reasonable solution to the problem of limited computer resources and a great number of students is to split the students into groups of fixed size. Each of the groups then has several fixed dates assigned and students are free to choose one of the groups. The advantage of this approach is that the group size is fixed and that the memberships and appointments are easily coordinated with other courses. The disadvantage is that individuals cannot change the dates associated with the group they belong to. If a student cannot attend at a set time, either individuals need to be allowed to change between groups on single occasions or changes of group membership as a whole have to be considered. This, however, eliminates much of the management simplicity which is among the original benefits of this method. Not taking this into account will result in students not being able to finish the course due to time-related reasons.

Another approach is to let the students choose individual appointments from a predetermined list of dates for each unit of exercises. More dates need to be listed than in the previous approach to accommodate all students, but on the other hand students have a large degree of freedom and flexibility in choosing their appointments.

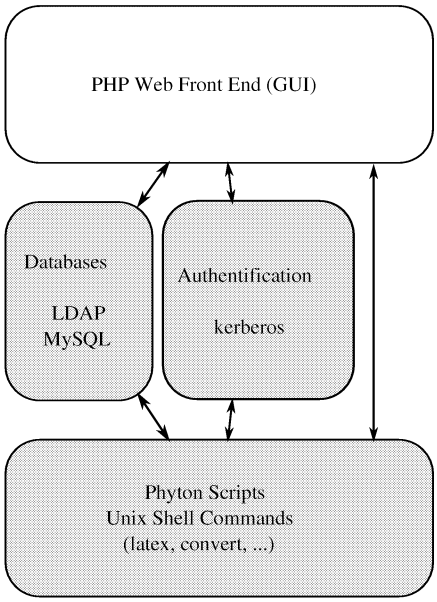


Figure 1: Based on an orthogonal software design the Labtool can be split into several modules with consistent interfaces.

An important fact related to social issues is the incorporation of the users. Whenever developing and implementing new management software, it turned out that acceptance among users is highest when

- they are involved in the development process and
- the software is fully backward compatible to any solution that may have been deployed before.

The integration of users in the whole development process not only increases acceptance, but also reduces the probability of erroneous use of the whole system.
The remaining issues listed in the requirements can be addressed by technical solutions which are presented in the subsequent sections.

THE DEVELOPED SYSTEM

The complete system consists of a laboratory cluster with several clients running basic services connected to two servers and the management software running on these servers. A brief overview of the structure of the management software is given in Figure 1 whereas an overview of the technical infrastructure is given in Figure 8.
The management software is composed of the following modules:

- a graphical user interface written in the dynamic web programming language `PHP` [7],
- user databases implemented in the Lightweight Directory Access Protocol `LDAP` [3],
- `python` scripts to produce lists in `LaTeX`,
- an authentication module implemented by the computer network authentication protocol `kerberos` [6] and
- a monitoring module to observe the status of system.

A very important part for the acceptance of such a system is the ease with which the most often used tasks can be executed. The whole system has not only to be easy to understand and use, but also has to be presented appealingly. The user has to be comfortable when working with the management software. We therefore put special emphasis to make all often used tasks available with as few clicks and inputs as possible.
Figure 2 shows the 'Enable' mask which allows to enable user accounts for students already registered for the current date as well as to join students who have not done so (register them for the current date and enable their user accounts). The input mask at the top provides an easy to use facility to tag students with specific unique identifications, names or registration numbers or parts thereof. Most of these functions can be controlled by various click functions, e.g., it is possible to enable the accounts of students with a singe click in the 'Enable' mask.
Also, we paid special attention on the concept of orthogonal application design: Each module can be used by itself or in combination with the graphical user interface (GUI), which furthermore makes it easy to modify, supplement or substitute single components.
In the following we present the main features of these modules. Authentication via `kerberos` is used to login to the Labtool. This has the advantage that we did not have to implement and maintain our own user management, but were able to use the data already provided by the central university system. Additionally many of the functions require the Labtool user to be known to the system, because only certain users have sufficient privileges to perform some tasks. We therefore implemented three groups of Labtool users:

- normal users, the instructors to hold the exercises,
- admin users, the managers of the course to additionally modify master data of the students, dates, and appointments, and
- system administrators able to monitor and restart services and the firewall.

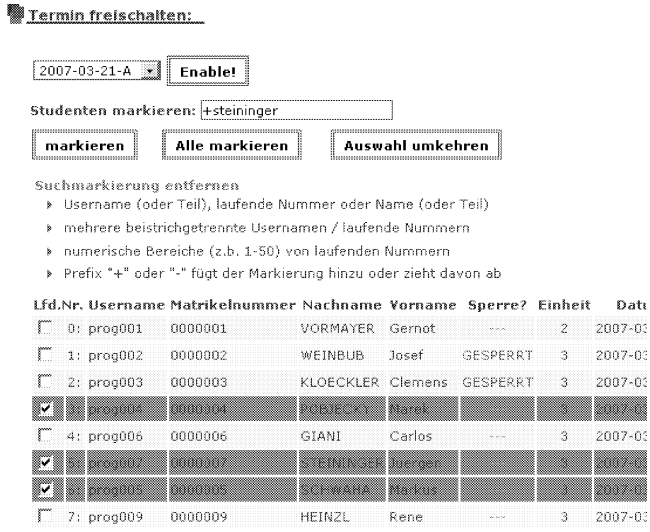


Figure 2: Registered students are automatically preselected. Additional students can be added to and removed from the selection using the input mask.

Another design decision was how to store user data in a database. The facts that the central university system supports `LDAP` queries against its students database, but also because the tree structure enforced by `LDAP` maps the structure of the course best and is optimized for these accesses, were strong arguments for ranking `LDAP` higher then other common solutions such as `MySQL` [8]. The opportunity to check the registration data such as names or registration numbers against the central database of the university is important when verifying the identities at the begin of exercises and much more when transmitting grades to the department of reporting. The most difficult time is during the beginning of the semester, when the courses already start but the term of registration has not yet expired. It is not always possible to verify the names and registration numbers of some of the students reliably during this period. A manual management system is therefore also available.
A student has several attributes, depicted in Figure 3, such as names, sex, registration number, and certain flags concerning his status at the course and at the system. All students have to attend each unit of the course. Therefore the student subscribes to each unit at a certain date thereby generating an appointment. Each appointment is then evaluated. If a student shifts the date of an appointment or is absent for some reason, another appointment is made available in the tree. Otherwise, if an appointment has a valid

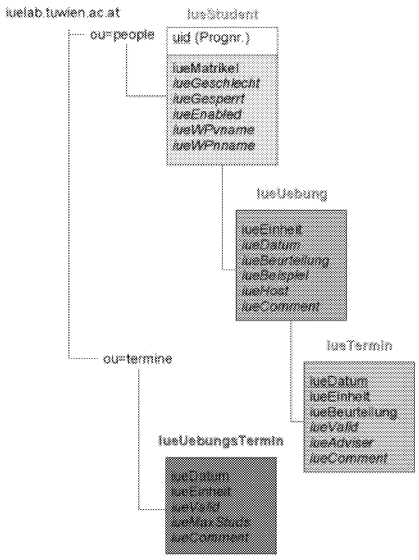


Figure 3: The LDAP-Tree shows objects and attributes we implemented. The attributes written in italics are optional.

grade (the student attended the exercise), the respective unit is evaluated. Figure 3 shows the student related information first, then the separate units, then the appointments, and finally the possible dates to subscribe for the units.

There are many reasons for keeping the possibility to print out reports of presence and success of a specific day or the overall list of the course. The most basic reason is to have a hard-copy backup in case the whole system crashes. Another reason would be the possibility to keep the course running, while the management software is temporary out of order. We already pointed out that acceptance also can be approved by keeping methods to easily downgrade to an older system due to the fact that people do not want to be tied to changes that cannot be undone.

For all these reasons and due to the concept of orthogonal application design we decided to implement `python` scripts and `LaTeX` templates to produce `postscript` files that can be stored and printed within and without the management software. Figure 4 shows an example for such a list for a single day.

LISTE PROGRAMMIEREN 2 (Ü3) 2007-03-21A

O	prog004	0000004	POBJECKY Marek
▽	prog005	0000005	SCHWAHA Markus
-	prog007	0000007	STEININGER Juergen
#	prog009	0000009	HEINZL Rene

Registered: 4
Missing: 1

Figure 4: The list of the day is produced with `python` scripts and `LaTeX`.

The graphical user interface front end is written in a `PHP` web application for many reasons:

- Every computer providing a web browser is able to control the Labtool,
- people are accustomed to web front ends,
- no necessity to implement a separate GUI, and
- `PHP` provides facilities to connect to all other modules (`LDAP`, Unix Shell Scripts, `kerberos`, `MySQL`, etc.).

For security and usability reasons the GUI is split in several sections (See Figure 5).

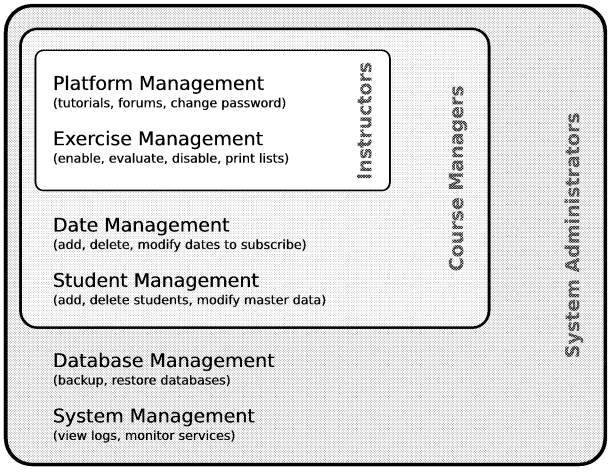


Figure 5: The GUI is divided in sections corresponding to the user groups.

The sections dealing with the platform specific tasks (tutorials, forum to post new ideas regarding the platform, password change functionality) and holding an exercise are available to all users. To further increase comfort and minimize needless output of paper we implemented a preview for all lists that may be printed. We use `ImageMagick` for the necessary conversions. Clicking a provided thumbnail opens a new window with the preview as shown in Figure 6.

The sections to control new dates to subscribe to and manage the students' master data are only available to managers of the course. They can add, delete, and modify the data of students, dates, and appointments. It is also possible for them to modify the grades of any student and any appointment in this section. The last part of the GUI is reserved for the system managers and provides logs, monitoring modules for the services and the firewall as well as the possibility to backup and restore the underlying databases.

One of the features to increase comfort for the user and the security of the system is to offer the possibility to easily undo changes or manage backups from different particular dates. This solves the problem of data loss in general and eases the handling of the whole system. A second reason for data loss is fragmentary transfer of input which can be avoided by redundancy.

Security is also threatened from outside of the system. Wherever computers have a high bandwith link to the world wide web, like at universities, the rates of attack can be considerable. Furthermore students attempt to get access to management systems in order to obtain information, change their grades, or try to get unfair assistance from other students or the Internet. For all of these reasons our Labtool is protected on the one side by a firewall and intrusion detection systems, and on the other side with additional database encryption in sensitive areas.

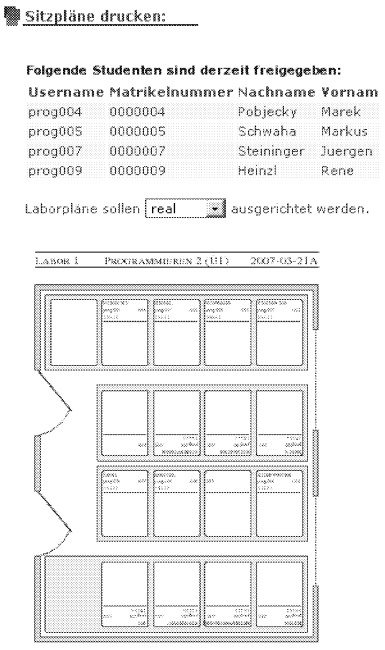


Figure 6: Clicking on the small lab maps opines a new window with the preview.

PROBLEMS AND SOLUTIONS

One of the the major problems we want to point out was the deficient realization of the `kerberos` functions in `PHP`. Because available documentation was badly lacking, our approach was to write our own `kerberos` library for `PHP` using the shell commands of `kerberos` and the `exec()` function of `PHP`.

Also, the system needs a great number of different services running, devices to be mounted, and systems to be synchronized. Therefore we not only implemented modules to monitor all of these services and states but also a kind of self healing mechanism. The monitoring modules ensure that system administrators can find sources of errors easily and quickly (see Figure 7).

The self healing modules try to avoid specific problems, such as:

- asynchronous system clocks of the clients or unreachable `NTP` service on the server,
- not mounted net devices on the clients or unreachable `NFS` service on the server,
- unreachable `kerberos`, `syslog` and `fcron` services regarding the system and
- unreachable `apache`, `LDAP`, `MySQL` and `cups` services regarding the Labtool

by periodically checking their status and attempting to restart the services, and synchronizing in case of errors.

The need to keep all system clocks of the cluster synchronized originates from the use of `kerberos`. It is due to the fact that `kerberos` uses timestamps as part of its authentication, the tolerance of clock skew lies within 300 seconds [6]. The situation gets even more difficult, when, for security reasons,



Figure 7: The current status of all important services can be monitored.

only the servers are able to connect to the Internet. The servers update their local clock from time servers outside the cluster and have to propagate this new time to all clients fast enough to stay within the interval of tolerance in order to keep authentication working. Whenever a client was offline, it first has to synchronize its time with the server before authenticate against `kerberos` is reliably possible.

To establish certain standards of security for the servers against threats from outside the lab cluster we implemented not only a firewall with `iptables` but also employ port scan detection (`psad, portsentry`) and intrusion detection tools (`samhain, aide`). Furthermore we installed the hardened kernel [5] sources and built a monolithic kernel to block as many routes for break-in attempts to the system as possible and alert the system administrators automatically via email.

On the side of the clients in the cluster we do not allow communication with hosts outside the specified range of IP addresses by not providing a default gateway. Due to the fact that `SHFS` is not a real distributed file system, its unavailability at boot time and because further utilities are needed in addition to the kernel modules for its deployment, we had to choose a solution from the following file systems:

- `AFS`, which uses only version 4 of `kerberos` (which makes it vulnerable to plain text attacks [2]) and the stateless UDP protocol (which makes it difficult to use with firewalls) and being only tagged as experimental in the kernel which we used.
- `CODA`, which has low speed of writing and deleting files, no mechanism to make files persistent other than closing the file (which makes it difficult to use for log-files and other files kept open for long periods of time), and requires that entire files have to be read [4] (which means that the cache must be at least as big as the biggest file will ever be).
- `NFS` version 4 [1], which uses the state-full TCP protocol instead of UDP, allows authentication of users via `kerberos` version 5, and is economic in its use of memory. In our experience it still suffers from a few stability issues and may even freeze the system, if any of the `NFS` related services fail.

To ensure minimal downtime of the system in case of a crash of a server we mirror all necessary services, such as:

- `NFS`, `kerberos`, `syslog` and `fcron` (system functionality)
- `apache` and `cups` (Labtool functionality)

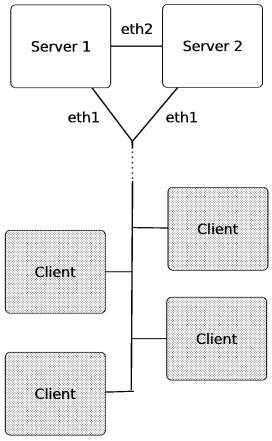


Figure 8: The cluster is connected via Gigabit Ethernet Connections.

and the following two databases:

- `LDAP` for the main databases
- `MySQL` for additional logging purposes

on a second server (see Figure 8). Furthermore all data from the students (i.e. home directories) and the Labtool (i.e. all lists) are backuped on a host outside the cluster every day.

OUTLOOK

We have several ideas how a system such as Labtool can be enhanced and extended and present them in the following. One possible extension is to implement methods that the success of students can be evaluated by the tool automatically. To this end it is necessary that the computer programs implemented by students have defined interfaces for input and output via the `main()` function. Having a catalog of stages with a defined interface in this manner, the Labtool could easily detect the stages already working. Currently the tasks of the instructors in our laboratories are not only to provide assistance to the students with their programming tasks, but also perform a final inspection of the students' programs and to evaluate their skills. Enhancing the software this way would decrease the amount of work of the instructors and increase objectivity of assessment at the same time.

Another improvement would be to handle the management of appointments completely within the Labtool. The advantages aside from saving work and time are first the lapse of transfer interfaces between different programs and therefore the omission of a source of error. Second the students would be able to do bilateral changes of appointments by themselves. A possible way to achieve this is to enable the students to send some kind of invitation to change an appointment such as the invitations sent by social software communities / social networking websites to either a specific student (for this the students must be able to see which student is registered for which date) or to all students registered for a specific date. The other student can then accept or decline this invitation. If a student accepts, the appointments are changed and all other students get a message that the invitation is recalled.

Implementing the feature this way would allow changes without being online at the same time or on a common interface, and no session keys would have to be exchanged.

CONCLUSION

We presented the development and implementation of a comprehensive management tool for computer courses. We thereby outlined the use of the orthogonal software design concepts to keep the modules of the system exchangeable. Special attention is paid to error tolerance, monitoring and controlling software as well as self healing mechanisms. The system had to be integrated into a running computer course with an emphasis on backward compatibility. Usability was always not only kept in mind, but was one of the paramount goals of the development. We also discussed problems that surfaced and possible ways to resolve them, so that an interested reader should be able to re-implement a similar system and to avoid most of the problems that typically arise.

REFERENCES

[1] *NFSv4*, 2005. `http://www.nfsv4.org/`.

[2] *Open AFS*, 2006. `http://openafs.org/`.

[3] *Open LDAP*, 2006. `http://www.openldap.org/`.

[4] *CODA Filesystem*, 2007. `http://www.coda.cs.cmu.edu/`.

[5] *Hardened Gentoo 2007.1*, 2007. `http://www.gentoo.org/proj/en/hardened/`.

[6] *Kerberos: The Network Authentication Protocol*, 2007. `http://web.mit.edu/kerberos/www/`.

[7] Achour, Betz, Dovgal, Lopes, Olson, Richter, Seguy, Vrana, et al. *PHP Manual*, 2006. `http://www.php.net/manual/en/`.

[8] MySQL AB. *MySQL 5.1 Reference Manual*, 2006. `http://dev.mysql.com/doc/refman/5.1/en/index.html`.

BIOGRAPHIES

RENÉ HEINZL studied electrical engineering at the Technical University Vienna. He joined the Institute for Microelectronics in November 2003, where he finished his doctoral degree in September 2007. His research interests include computational science, high performance programming techniques, solid modeling, scientific visualization for TCAD.

GEORG MACH was born in Vienna, Austria, in 1979. He studies electrical engineering at the Technical University Vienna. He joined the Institut für Mikroelektronik in June 2003, where he is currently working on his diploma thesis.

PHILIPP SCHWAHA studied electrical engineering at the Technical University Vienna. He joined the Institute for Microelectronics in June 2004, where he is currently working on his doctoral degree. His research activities include circuit and device simulation, device modeling, and software development.

SIEGFRIED SELBERHERR received the doctoral degree in technical sciences from the Technical University Vienna in 1981. Since that time he has been with the Technical University Vienna as professor. Dr. Selberherr has been holding the "venia docendi" on "Computer-Aided Design" since 1984. As of 1988 he has been chair professor of the Institut für Mikroelektronik. From 1998 to 2005 he served as Dean of the "Fakultät für Elektrotechnik und Informationstechnik" at the Technical University Vienna. His current topics of interest are modeling and simulation of problems for microelectronics engineering.

Electro-Biological Simulation using a Web Front-End

P. Schwaha, R. Heinzl, G. Mach, C. Pogoreutz, S. Fister, and S. Selberherr

Institute for Microelectronics, TU Wien
Gußhausstraße 27-29/E360, A-1040 Vienna, Austria
E-mail: {schwaha | heinzl | mach | selberherr}@iue.tuwien.ac.at

KEYWORDS
Simulation, Biology, Web Application, Linux

ABSTRACT
We have developed a web application for the simulation in the field of electro-biology, particularly, for the analysis of a weakly electrically active fish. We investigated the charge relaxation due to the electric activity of the fish. The governing relaxation equation is discretized with the finite volume method. The discretized equations are then implemented using a generic scientific simulation environment. To ensure high responsiveness and scalability we also included means to distribute the calculations to computers on a network.

INTRODUCTION

The pursuit and the acquisition of new insights is common to all sciences. However, the methodologies and tools utilized to advance on the goal are quite different, due to the very different natures of the various scholarly disciplines. Simulation is a technique that has been applied very successfully in several fields of science, such as physics, and has even spawned its own branch of research: scientific computing. While simulations have proven to be very important to understand the various phenomena in nature, some areas of research have all but neglected this technique.

In order provide a means to quickly implement even complex algorithms directly in C++ we developed the scientific simulation environment (GSSE) [5, 6, 8, 13] which not only provides high performance but also takes into account the often neglected topic of software quality. Although its roots can be traced to satisfy the needs of technology computer aided design (TCAD) [4, 11, 12, 14], its application is not limited to this field of research. Investigations concerning this during several student projects have been very positive [10].

One of these projects, that dealt with electric phenomena in biological organisms, also revealed that high performance is not the only important factor when cooperating with scientists with very different backgrounds. In this context the benefit of appropriate visualization should also not be underestimated [2], as it often determines how well the simulation results can be interpreted by the target audience.

The goal therefore was to provide an user interface that is easily accessible by people from different fields of research with a minimum of effort. By following modern application design rules, that components should be implemented orthogonally [7] it is made possible that the underlying simulation remains unmodified, which reduces sources of errors.

Because access to the world wide web has become all but ubiquitous in the last years, it is a promising choice for a user interface. Most people are familiar with the interfaces and semantics established in web pages and HTML forms. It can be therefore viewed as common ground for people of scientifically different backgrounds. From this point of view web pages are well suited as an interface to applications that bridge the gap between different branches of science.

THE SUBJECT UNDER INVESTIGATION

Electric phenomena are common in biological organisms such as the discharges within the nervous system, but usually remain within a small scale. In some organisms, however, the electric phenomena take a more prominent role. Some species of fish, such as *Gnathonemus petersii* from the family of Mormyridae [3, 15, 16], use them for detection of their prey and for communication among their own kind.

The nose region of *G. petersii* is modified to form a chin or snout like form. The shape representing the fish in the simulations is given in Fig. 1. The snouts possess very sensitive receptor cells which help to find small prey in the mud.

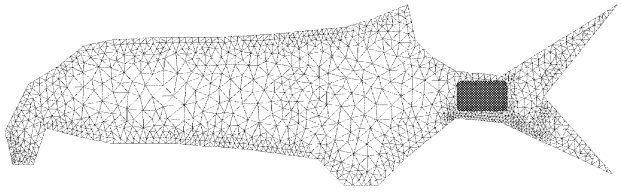


Figure 1: A triangulated model of an electrically active fish. The position of the electrically active organ is marked in red.

The goal of our simulations is to analyze the electric field generated by the fish, when different objects are introduced in the vicinity of the fish. This is difficult to obtain reliably with live specimen. The web application is described in the following section.

THE WEB APPLICATION

A web application as a front end to the simulation environment has to fulfill several tasks:

- Choose and modify the simulation parameters.
- Start the simulation with the selected parameters.
- Process the results of the simulation corresponding to the parameters and present them to the user.

A web application has to accomplish these tasks without interweaving the simulation itself with the presentation and control parts. In addition the strain on the web server itself should be kept to a minimum, in order to ensure low latencies. A model of such an architecture is shown in Fig. 2.

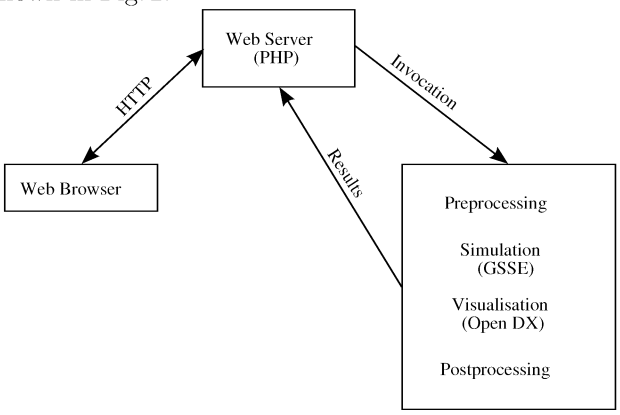


Figure 2: Overview of the web application's architecture.

The user contacts the web server using a standard web browser. After some preprocessing, the web server invokes the simulation. This is the point at which the simulation can be easily delegated to another computer. After the simulation has finished the post processing module generates appropriate output which the web server sends back to the user's web browser. The whole process has to be accomplished in a time frame of seconds in order to remain comparable to that of standard web browsing. Therefore, both, the simulation and the generation of a response, need to be implemented in a performance conscious way.

Our Implementation

Our implementation of the web application can be accessed under `http://webapplication.gsse.at/` to give the reader the opportunity for experimentation. The user can choose from three different geometric constellations of the fish and an object (thing) as shown in Fig. 4. Three different mesh resolutions are available with approximate simulation times given for each of them. Apart from the geometric configuration it is of course also possible to adjust the parameters of the simulation using HTML forms. In our case this includes the conductivities and permittivities of the various simulated objects (thing, fish, skin, water) and the rate of charge separation (production rate) inside the fish.

As has already been stated, the communication time between the simulating node and the web server must be appropriately short. In our setup the communication time was kept below 0.2 seconds when using a separate computer for simulation. With GSSE the simulation times themselves are kept in the same order of magnitude, at least for the coarser meshes. We have thereby

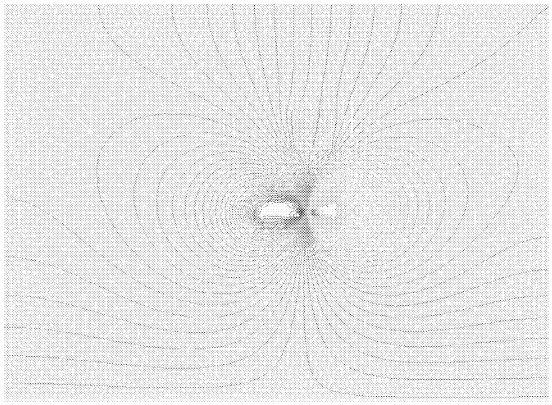


Figure 3: The results of a simulation performed with a very fine mesh.

ensured the orthogonality of the core simulation and the presentation of the results. Furthermore, we do not need to specially adapt the simulation software in any way to be used from the web application. This not only reduces maintenance, but also removes an unnecessary source of errors.

We use OpenDX [9] to generate images from the results of the calculation, because of its scripting capabilities. A resulting image is shown in Fig. 3.

The parameters entered into the web application and passed to the simulation are recorded along with the results. It is thereby possible to efficiently cache the results which have already been requested at least once. When the same parameters are entered for another run of the simulation, the calculation is skipped and the cached result is returned. This simple mechanism helps to conserve computer resources.

Additional details concerning the development are available at `http://tutorial.gsse.at/webapplication`. The modeling and implementation of the electric phenomena is discussed in the next section.

SIMULATION ENVIRONMENT

To realize a simulation we require a model of the physical phenomenon that can then be implemented. Therefore we first have to establish the theoretical foundations of our simulation.

We model the electrical phenomena due to the fish by a relaxation equation which can be directly derived from Maxwell's equations in a quasi-electrostatic case. It takes the following shape:

$$\partial_t \operatorname{div}(\varepsilon \operatorname{grad} \Psi) + \operatorname{div}(\gamma \operatorname{grad} \Psi) = P \qquad (1)$$

Here Ψ is the electrostatic potential, while ε and γ

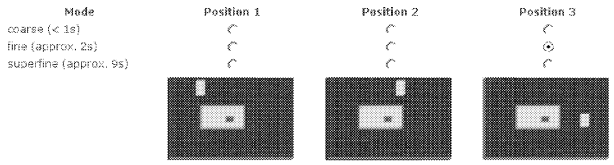


Figure 4: Different geometrical placement and meshes.

describe the permittivity and the conductivity respectively. The separation of charge, which is actively performed by the fish, is modeled by the inhomogeneity P. Equation 1 has to be solved numerically, because a closed form analytical solution does not exist for the general case.

GSSE

The GSSE has been developed in order to provide a reusable high performance framework for scientific computing at a high semantic level with C++.
Equation 1 is discretized using the finite volume discretization scheme [1] and is then implemented using the facilities offered by the GSSE. The following snippet of code, which already represents all relevant parts of the final application and can be used for arbitrary spatial dimension, illustrates the expressive power of the GSSE:

```
equation = sum<vertex_edge>
[ sum<edge_vertex>
  [ linearize(psi) ]
  * (area / dist) * (gamma * delta + eps)
] + vol * ((P * deltat) + rho)
```

All error prone details related to traversal or data access operations are taken care of by the GSSE, therefore, it can be used on structured as well as unstructured spatial discretizations called grids or meshes respectively.

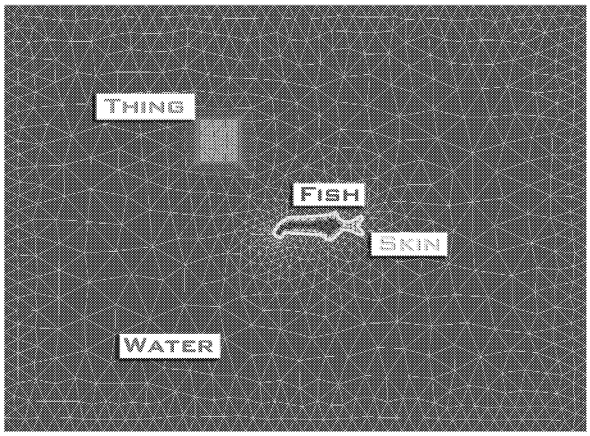


Figure 5: Simulation setting

The simulation domain is divided into several parts as shown in Fig. 5. The parts include the fish itself, its skin, which serves as insulation, the water the fish lives in, and an object which represents either an inanimate object or prey. The parameters of each part can be adjusted separately.
While the shown specification of the simulation retains a lot of high level semantics and yields a highly performant executable, an appropriate interface must also be provided to make it as accessible as possible.

CONCLUSION

We have shown how we used a web application as an interface to a simulation composed of orthogonal components. The core is implemented using the high performance GSSE, ensuring a minimum of runtime of the simulation itself. A caching mechanism helps to further conserve computational resources and enables fast responses. Our modular and orthogonal design furthermore enables an outsourcing of calculations from the web server. The ease of use of a web application hides the complexities and concentrates on presenting valid results and thereby makes the simulation more accessible to non-domain experts.

REFERENCES

[1] T. Barth and M. Ohlberger. Finite Volume Methods: Foundation and Analysis, 2004.

[2] W. Benger, G. Ritter, and R. Heinzl. The Concepts of VISH. In *Proc. of the 4th High-End Visualization Workshop*, pages 28–41, Tyrol, Austria, June 18–22 2007.

[3] Josef Dudel, Randolf Menzel, and Robert F. Schmidt. *Neurowissenschaft. Vom Molekül zur Kognition.* Springer, Berlin, 2001.

[4] T. Grasser (Editor). *Advanced Device Modeling and Simulation.* World Scientific Pub Co Inc, 2003.

[5] R. Heinzl. *Concepts for Scientific Computing.* Dissertation, Technische Universität Wien, 2007.

[6] R. Heinzl, M. Spevak, P. Schwaha, and T. Grasser. A High Performance Generic Scientific Simulation Environment. In *Proc. of the PARA Conf.*, page 61, Umea, Sweden, June 2006.

[7] R. Heinzl, M. Spevak, P. Schwaha, and T. Grasser. Concepts for High Performance Generic Scientific Computing. In *Proc. of the 5th MATHMOD*, volume 1, Vienna, Austria, February 2006.

[8] R. Heinzl, M. Spevak, P. Schwaha, and S. Selberherr. A Generic Topology Library. In *Proc. of the Object-Oriented Programming Systems, Languages, and Applications Conf.*, pages 85–93, Portland, OR, USA, October 2006.

[9] IBM Corporation, Yorktown Heights, NY, USA. *IBM Visualization Data Explorer*, third edition, February 1993.

[10] IμE. *IμE Summer of Code 2006.* Institut für Mikroelektronik, Technische Universität Wien, Austria, 2006. `http://www.summerofcode.at/`.

[11] C. Jungemann and B. Meinerzhagen. *Hierarchical Device Simulation. The Monte Carlo Perspective.* Springer, 2003.

[12] R. Sabelka and S. Selberherr. SAP — A Program Package for Three-Dimensional Interconnect Simulation. In *Proc. Intl. Interconnect Techn. Conf.*, pages 250–252, Burlingame, California, June 1998.

[13] P. Schwaha, R. Heinzl, M. Spevak, and T. Grasser. Advanced Equation Processing for TCAD. In *Proc. of the PARA Conf.*, page 64, Umea, Sweden, June 2006.

[14] S. Selberherr. *Analysis and Simulation of Semiconductor Devices.* Springer, Wien–New York, 1984.

[15] G. von der Emde and S. Fetz. Distance, Shape and More: Recognition of Object Features during Active Electrolocation in a Weakly Electric Fish. *J. Exp. Biology*, 210(17):3082–3095, 2007.

[16] Wilfried Westheide and Reinhard Rieger. *Spezielle Zoologie. Teil 2: Wirbel- oder Schädeltiere.* Elsevier, 2003.

SIMULATION IN EDUCATION

INTERACTIVE REFINEMENT OF A MATERIAL FLOW SIMULATION MODEL BY COMPARING MULTIPLE SIMULATION RUNS IN ONE 3D ENVIRONMENT

Matthias Fischer[†]
Christoph Laroque[*]
Daniel Huber[*]
Jens Krokowski[†]
Bengt Mueck[‡]
Michael Kortenjan[†]
Mark Aufenanger[*]
Wilhelm Dangelmaier[*]

[†]Algorithms and Complexity
Heinz Nixdorf Institute
University of Paderborn
Fuerstenallee 11, 33102 Paderborn
Germany

[*]Business Computing, esp. CIM
Heinz Nixdorf Institute
University of Paderborn
Fuerstenallee 11, 33102 Paderborn
Germany

[‡]Department MuT
University of Applied Science
Osnabrück
Am Wall Sued 16, 49808 Lingen
Germany

ABSTRACT

The validation of material flow models as well as the selection of promising strategies for the generation of a successful experiment plan is a time-consuming process. A new approach is presented, which supports the simulation expert in his working process by giving him the opportunity to modify the simulated simulation run and afterwards compare the effects of his modification with the original setting, online and in one user interface, implemented by switching the visualizations between the simulation runs or opening up to 5 parallel 3D windows. The method developed therefore clones existing simulation runs online and allows the navigation within these existing simulation runs. The method has been implemented and is validated by a test model, which describes in detail the new working process of a modeler. New research questions are derived from this work, which will define following working steps.

1 INTRODUCTION

Today's state of art simulation tools, such as Flexsim, Automod, eM-PLANT, SIMUL8, Quest3D, and others, provide 3D visualization of the simulation models. Modeling and simulation can be done completely in 3D. Even during simulation, extending parameters and changing model logic can be done "on the fly". The user can manipulate the simulation model during the simulation run and an interactive analysis of the simulation is offered. 3D visualization offers benefits in intuitive process understanding and result communication (Kamat and Martinez 2000).

Recent, results from surveying simulation practitioners from industry who have used 2D or 3D software applications for discrete-event simulation show that many remain cautious, guarding against possible exaggeration of the claimed "3D benefits" (Akpan and Brooks 2005a).

The analysis of several sequential simulations remains difficult because of the missing direct comparison. Have the changes really enhanced the performance of the system? Because the unmodified system does not exist, the user has no possibility to compare the modified system with the unmodified system.

Our approach is to run *multiple similar simulations* in parallel. The simulation runs differ in the distinct parameter values of the model. Because of distinct parameter values, the simulation runs might show distinct behavior, e.g., they differ in the number of packets that are moving across a conveyor belt.

Executing the simulation in parallel is not enough: a small set of the simulation runs (in our implemented system five) can be directly seen in the 3D visualization simultaneously on one screen. A direct comparison is possible and the user can analyze the differences.

Further interactive modifications can be done by a "*pausing-cloning-modification*" process:

- first the user pauses all simulations,
- afterwards a single simulation is selected for cloning,
- after cloning, the user modifies parameters of the cloned simulation run.

After the modification of the simulation run, all simulations proceed and both, the unmodified and the modified, i.e., the cloned simulation, are rendered simultaneously, so that the user can see the differences in two separate 3D windows on the same screen.

In Section 2 we discuss related work in simulations with 3D visualizations and parallel and distributed simu-

lation. In Section 3 we present a conceptual overview of the system. We present the architecture and utilization of the system in Section 4. We discuss some derived research questions in Section 5 and conclude our ideas in Section 6.

2 RELATED WORK

Our simulation system consists of a walkthrough system that renders multiple simulations runs on a single screen. The user can clone a single simulation run, modify parameters, and observe the 3D visualization of the cloned and modified simulation run. Therefore, first we review 3D visualization of simulation with 3D support, second we briefly discuss distributed simulations and cloning of simulations.

2.1 Interactive Simulations with 3D Visualization

The 3D visualization of simulated processes has successfully found its way into discrete simulation technology for the last decade (Barnes 1997); before the discrete simulators rendered the model in a 2D fashion with icons and lines (Hlupic 2000). Typical arguments for 3D visualization of a simulation modes are a better understanding of problem by all stakeholders and the easier identification of errors (Kamat and Martinez 2000), e.g., the detection of accumulations of packets on vertically arranged conveyor belts.

Most of today's discrete simulation systems (e.g., Quest, Automod, and others) already provide integrated 3D visualizations. In these systems, simulation and rendering running on the same machine. For performance reasons, simulation and rendering can run on different machines and are bidirectionally coupled with each other (Strassburger et al. 2005, Fischer et al. 2005). When coupling simulation with VR on different machines, the synchronization of both tools, i.e. the time advance in both tools, must be coordinated (Strassburger et al. 2005). Not only virtual reality is an application of 3D visualization, but also augmented reality based simulations (Behzadan and Kamat 2005).

Recent results of experimental studies that tested the impacts of Virtual Reality (VR) on Discrete-Event Simulation show that it is easier and faster to spot errors in 3D/VR model than in 2D (Akpan and Brooks 2005b). Despite the advantages of 3D visualization, many remain cautious (Akpan and Brooks 2005a) because of a long learning curve for 3D software.

We think the features of the 3D visualization are crucial to get a benefit. Our contribution in this paper is a new 3D visualization that supports the modeler by allowing comparison of settings before and after model modification.

2.2 Distributed Simulation and Cloning

To overcome the computation and memory limitations of single processor architectures, parallel and distributed execution of discrete event simulations is used to speedup the computation. One approach is the concurrent execution of a single simulation run in a multi-processor architecture. Another technique is to execute multiple runs of independent simulations in parallel on a set of concurrent CPUs. Every simulation run is executed from the beginning to the end on the same CPU. The single independent simulation results are collected for correct and statistically significant analysis. (Banks et al. 1996, Fujimoto 1990, Kalyan 2006).

We use a loosely coupled PC cluster for our simulation and visualization system. Rendering and simulation runs are executed simultaneously on concurrent CPUs. The user switches between the simulation runs in order to observe the progress and investigate the differences.

Cloning of parallel simulations is designed to analyze alternative scenarios of a distributed simulation concurrently within the same execution session. The goal is to reduce the execution time for evaluating different scenarios by avoiding repeated computation. A simulation is cloned on an event for a particular set of logical processes, creating new clones (Hybinette and Fujimoto 2001). Just-in-time cloning mechanisms (Hybinette 2004) avoid unnecessary cloning of logical processes. Schulze et al. (1999) introduced a cloning approach based on HLA to extend the flexibility of system composition during runtime, and proposed the cloning of federates at runtime. Forecast functionality is provided by their approach via the parallel scenario management of different time axes. Chen et al. (2005) improves the cloning of HLA-based distributed simulations by cloning mechanisms and infrastructures that ensure state consistency and support user transparency and reusability of federate codes.

Our system provides cloning of simulations as means for running multiple similar simulations in parallel. However, unlike (Hybinette 2004), our goal is not to share computations. Some of the cloned simulations are modified by manipulation of parameters. The modified simulations are rendered and visualized in parallel in a 3D walkthrough system. Our cloning implementation uses ideas of the approach of Schulze et al. (1999). We store all relevant elements of the simulation model, such as object instances and other internal states of the model.

3 CONCEPT

Based on the fundamental idea of improving and advancing the workflow of a simulation expert in the area of the material flow simulation, the concept presented and its implementation uses an interactive material flow simulator with 3D-visualization not only for the representation of one single simulation run, but to show noticeable dif-

ferences and similarities between multiple simulation runs in one virtual environment.

The development process of a simulation model by a simulation expert can be enhanced if the interactive modifications and their consequences can be compared to the original setting during the continuous execution of the simulation run. Variants in the simulation model, as well as different control strategies can be derived from the original setting in one working step and without restarting the simulation run.

3.1 Overview

The main goal of this work is to enhance and accelerate the simulation expert's work in designing and validating a material flow simulation model. Moreover, the user will be enabled to design better model variants and control strategies based on the original setting. Therefore, in a first step, an interactive environment is needed which generally allows the modification the system modeled and its dynamics during its execution and follows the effects of the modification. In order to allow any analysis of which behavioral changes result restrictedly from the interaction and which were already defined in the original model, the user needs a possibility to compare the original simulation run with a configuration which includes his modifications. Thereby, this comparison allows judgement of the modifications aimed at the goal of model improvement.

We want to achieve This goal by the use of a combination of interactive modifications, the use of a cloning mechanism to create parallel simulation runs without restarting the simulation, and the representation and control in one single 3D-visualization system.

3.2 Workflow

The implementation of the ideas presented also leads to changes of the user's workflow in order to improve and accelerate his work. The resulting workflow and its consequences for the implementation of the system architecture is described in this paragraph. So how does it look?

The user develops and designs a first draft of a simulation model and simulates it in order to check the modeled behavior. Thereby, a 3D-visualization in a virtual environment helps him to easily recognize the system dynamics and compare them to the goals of the simulation study. The user has to validate the original simulation model in respect to the experimental frame or change the behavior of the modeled system. Normally, this leads directly to the loss of the original setting, so that the following dynamics of the simulation run cannot be divided into those which result directly from the modification and those which were already defined in the model. When using common simulation tools, the user has to stop the simulation, modify the original model and restart the simulation, which leads to a time-consuming work process.

The process of an interactive modification is accelerated by a cloning mechanism (see Figure 1), which is started after a user-defined set of interactions. This mechanism clones the original simulation run with its current state and assigns the set of interactions only to the copy of the simulation run. This way, the original setting is saved and the consequences of the interactions can be compared directly and without restarting the simulation run.

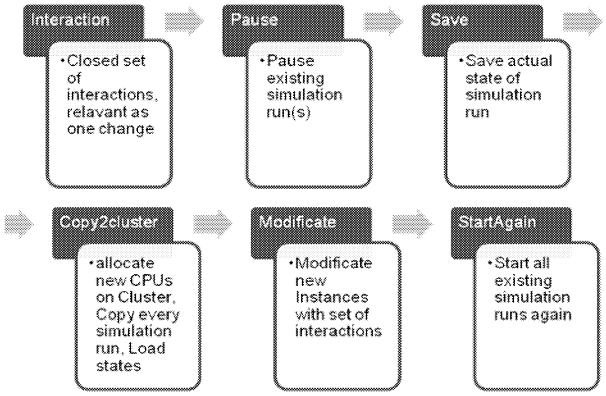


Figure 1: Cloning Process

By the additional use of a visualization system, which allows switching between or tile the 3D-visualization of each existing simulation run (more than one set of interactions clearly leads to several simulation runs), the direct comparison of different simulation runs is allowed.

As a consequence of this workflow, the need for a platform for several simulation runs and their visualizations arises which allows the execution of 10 or more simulation runs, even if the simulation model itself is very complex. During the implementation, a system architecture was to be developed as first step, which supports this working process of a simulation expert. An effective navigation in a big set of simulation runs is difficult, but can be improved, by designing a visualization method, which allows the viewing and control of the existing simulation runs. In this approach, an "experiment tree" was developed (see Figure 2) , which covers this method and supports the user in finding the right simulation runs for a meaningful comparison.

The computational power needed cannot be restricted to the simulation runs exclusively because navigation within the experiment tree leads to the switching of the visualizations, or, in case of multiple rendering windows, to the visualization of several simulation runs in one user interface. Therefore, the existing computational power has to be balanced between the simulation runs and their rendering as well as with the rendering itself. During the implementation of the system architecture, load-balancing problems are also solved.

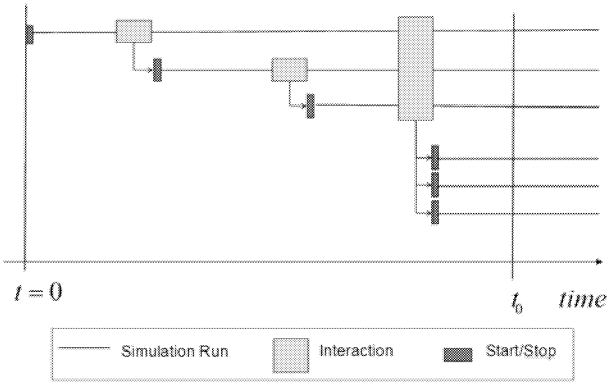


Figure 2: Experiment Tree

3.3 Advantages

By pursuing this new user workflow, the simulation expert is able to improve his understanding of the process and has the opportunity to compare and, therefore, directly judge the consequences of his modifications in a running simulation model. Using this new method of analysis, he is able to evaluate if his modifications indeed lead to a better performance of the simulation model or one of its variants. Different control strategies can be compared in one view, so that only those, which promise maximum success are to be ensured through simulation experiments. Validation and improvements during the modeling phase of a simulation experiment can be made faster, safer and, as a result, with less time and cost.

4 UTILIZATION IN MODEL ANALYSIS

In this section, the workflow in the system designed is demonstrated. Starting with the description of a small model, the usage of the system and the techniques to compare different system configurations are presented.

4.1 The Test Model

The model consists of two workers doing quality control on three different product types. The quality tests demand manual work at the benches B1-4. Worker W1 works at benches B1 and B2; worker W2 works at benches B3 and B4. Product P2 is an uncommon product, but when pieces arrive at B1, they are of top priority and have to be tested immediately. P3 is incoming at a constant rate, but it is not very important. When the input rate of P2 is normal, W1 can test products P1 and P2 alone. To absorb peaks in the input rate of P2, a distributor D can divide P1 between B2 and B3. If so, W2 has to test P1 on bench B3 next to his normal work.
In the following paragraphs, this model is used in an experiment to find a strategy for distributing P1 between the two workers in case of a peak in the input rate of P2. It is assumed that there is a delay time until the realization of the peak and the start of distributon of P1. So there will be a queue of P1 in front of B2 before the distribution begins. The distribution should assure that there are not many waiting pieces of P1 and W2 is only involved in testing P1 at a minimal level.

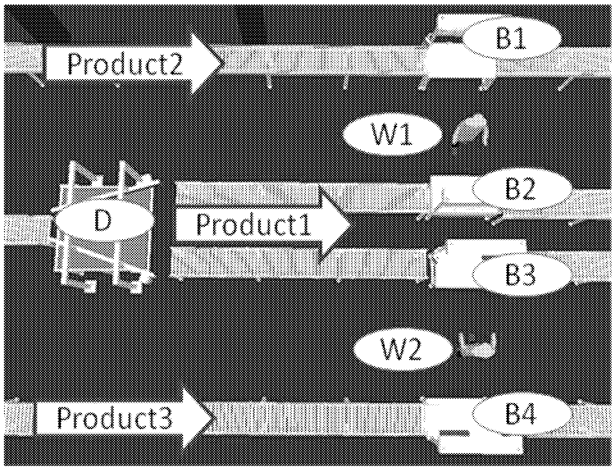


Figure 3: Test Model Overview

4.2 Step 1: Creating new Simulations with varying Parameterization by Cloning

Assume the user wants to compare systems with distribution rates between B2 and B3 by 1:0 (only B2), 1:2 and 1:1 (equal distribution). The rate of 1:0 is applied in the original system, thus two new simulations have to be created. To create a new clone, the user has to:

1. Pause the simulation
2. Apply the parameter changes on the simulation entities. In this case setting the distribution rate to 1:2. (Clone 1 in Figure 4)
3. Initiate the cloning process
4. Repeat with 2. to set the distribution rate to 1:1. (Clone 2 in Figure 4)
5. Unpause the simulations

The actual cloning in 3. is performed by:

1. Saving the unchanged simulation
2. Creating a new simulation kernel and renderer
3. Loading the saved simulation without canceling the original one
4. Applying the parameter changes to the cloned simulation

4.3 Step 2: Comparing the System Configurations

After cloning, three simulation runs exist. Each one is visualized in a window (Figure 4). The model performance can be analyzed by viewing all windows tiled next to

each other. In this example, it can be seen that the number of pieces waiting in front of B2 increases when the distribution rate is 1:0. The rate of 1:2 over compensates the effect and 1:1 generates a equal queue length. Thus setting the rate to 1:1 in case of a peak seems to be a good strategy.

Comparing three different systems is not difficult. But the number of simulations can increase rapidly when analyzing several strategies simultaneously. One scenario would be to find an optimal distribution rate if there is an option to test only every second or third piece. This would result in 3(distribution rate) x 3(test volume) =9 simulations.

4.4 Step 3: Navigating the Experiment Tree

To analyze an experiment with more than a hand full of simulations, more efficient techniques are necessary. To organize all simulations created, trace changes, or stop single simulation runs, an *Experiment Tree* was created (Figure 4, top left). Every simulation run is a line extending over the time axis. When clones are generated, new lines are created indicating which simulation is the origin. When picking one line, the corresponding simulation is visualized in the main window. If multiple lines are picked, multiple simulations are visualized, one in the main window, the others in the smaller ones. .

5 SYSTEM ARCHITECTURE

In this section, the system architecture capable of computing and visualizing parallel replications of simulations is presented.

The number of simulations created by cloning can increase rapidly, thus implementation on a simple computer is not feasible and the system has to be built to run on a computer cluster. The system architecture is shown in Figure 5.

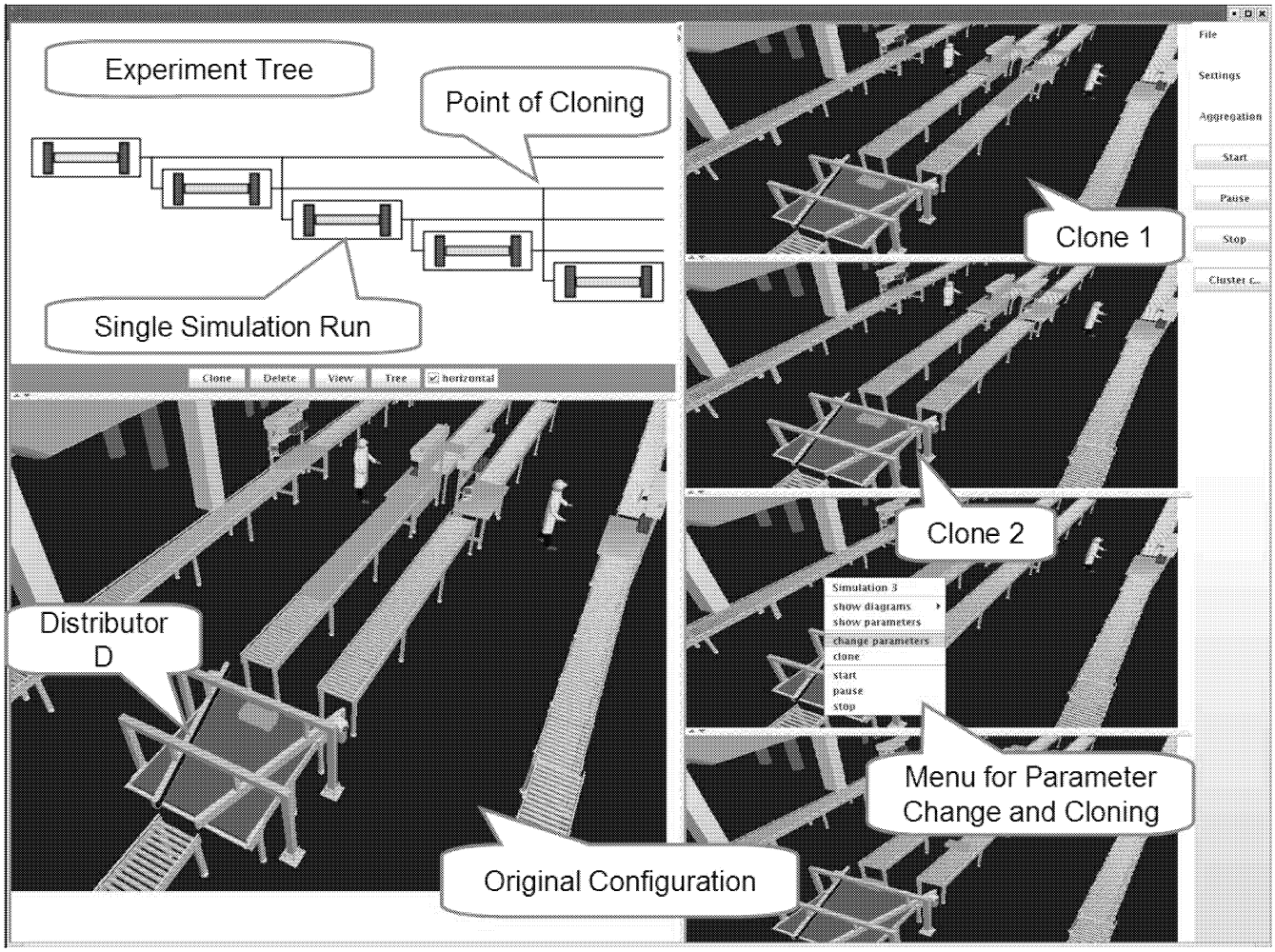


Figure 4:Graphical User Interface

A number of n nodes (SR) render the static objects of a simulation model. These could be conveyor belts, machines, etc. Because static objects are the same in all cloned simulations and the view frustum in every visualization window is the same, the static objects have to be rendered only once, no matter how many simulations are compared. To allow efficient parallel rendering of the static objects, a load balancing is done between the (SR) nodes. Therefore the view frustum is tiled in rectangles with an equal amount of polygons.

The dynamic objects, like tokens or forklifts, can differ in every cloned simulation. With no difference in dynamic objects, two simulations are exact copies. Thus one image of the dynamic objects per simulation has to be generated. One node (D) per simulation is allocated for rendering the dynamic objects (DR) as well as computing the simulation (Sim). For computing the simulation, the simulator d³FACT is used (Dangelmaier et al. 2005). For each clone created, one new node (D) has to be allocated. All rendered images are composed into one final image on the Mother Node. The image composition in the Mother Node first composes the static scene by joining the images send by SRs. Secondly, the image of all visible dynamic objects of one simulation is added. By using the Z-Buffer, occlusion of dynamic objects by static ones is achieved.

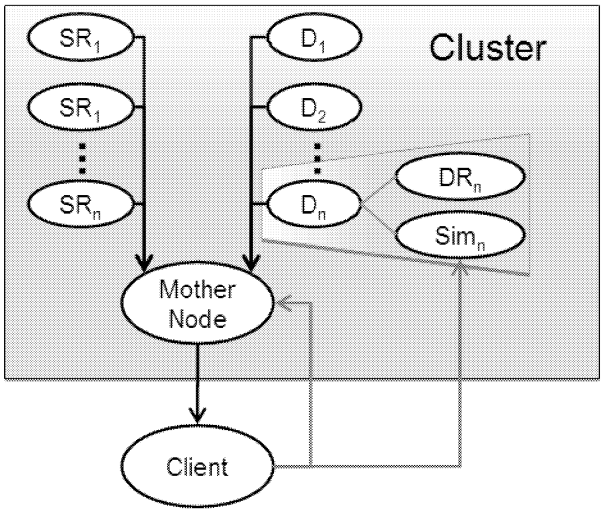


Figure 5: System Architecture

This final image is sent via Ethernet to a client computer exterior to the cluster. The in-cluster communication is transferred via Infiniband for higher performance. Because the Mother Node composes the image shown in the client, the client has to send information about the view frustum and the simulations to be visualized. For cloning and simulation data acquisition, the client can communicate directly with the simulations on the nodes (D).

To allow efficient comparison of simulations, the simulation times of all clones have to be synchronized. But because there is no interaction between simulations, the accuracy of this synchronization does not have to be very high. Synchronization is done by giving the simulations a time interval of simulation time for computation and waiting until all simulations have reached the end of the interval. After that, a new interval is sent to the simulations.

6 DERIVED RESEARCH QUESTIONS

During the implementation of the ideas presented, several further ideas by strictly following the user's demands arise. These ideas define new research areas still to be realized. The most important are described below.

6.1 Aggregation of Multiple Simulation Runs in a Single Virtual Scene

By working with the system architecture designed, the comparison of different simulation runs gets more and more complex when the number of interactions increases. After some interactions, either the user gets more and more confused by the often necessary switches, or the number of parallel windows is so high that the differences within the visualized simulation runs are not obvious any more.

As a solution, the presentation of several different simulation runs in one common virtual scene is imagined. Special work is still to be done to face those data, which is to be emphasized for an easy and fast analysis. As a first step, a scheme is to be defined which covers possible differences between these simulation runs, e.g. static differences (number of machines) or dynamic differences (position of a forklift or a package on the conveyor). The scheme designed also has to cover possibilities for the visualization of different internal states, e.g., the current state of a machine or its workload. For each element, an adequate type of visualization is to be identified. For example the question arises, how to visualize the aggregation of the forklifts' routes.

6.2 Visualization of Stochastic Processes

The material flow simulation is furthermore a stochastic process, which directly leads to the insight that the origin simulation, its dynamic behavior, and its visualization is not necessarily representative for the system mapped. The origin setting therefore should not consist of a single simulation run, but should cover a configuration of distributed simulation experiments, where every single simulation run just differs from the others in the start values of the stochastic parameters. The cloning process gets more complex this way, because several simulation runs are to be paused, saved and distributed within the computing cluster. This complicates the balancing problems mentioned and increases the computing power necessary.

Nevertheless, the number of simulation runs increase in a more relevant way with interactions: in the current system setting, two sets of interactions lead to up to 4 simulation runs. Starting with a configuration of 10 distributed simulation runs, the same interactions would lead to 40 distributed simulation runs, which sooner or later will cover all available computing power in the cluster. Therefore, new methods must be identified by which the number of simulation runs can be reduced by selecting only relevant simulation runs and thereby thinning out the entire experiment tree, which itself has to be improved for a efficient navigation within such a complex set of simulation runs.

But the intended solution also offers additional possibilities for an efficient analysis by the user. By regarding the entire configuration of more or less 10 simulation runs, the system can offer the user not only a specific parameter value, but also an aggregated range of existing values within this configuration additionally enriched by statistically analyses, e.g. average, median, standard deviation, etc. This allows the correct classification of the visualized scene and gives more detailed information about the system behavior.

7 CONCLUSION

In order to improve the workflow of a simulation expert dealing with the material flow simulation, a new approach has been introduced which allows the direct comparison of simulation runs in one user environment. By cloning a simulation run after a set of interactions, the validation and control process of the modeler can be accelerated because the effect of the interaction can be judged directly. The system therefore clones the running simulation and refines just the clone with the set of interactions. The method has been implemented and validated by a test model, which leads to new research questions. After a large number of interactions, the switching between the resulting simulation runs gets more and more complex. New methods need to be found to visualize different simulation runs in one window. Furthermore, the starting simulation run is not necessarily representative for the mapped system. To avoid this source of error, the system architecture is to be enlarged, so that the starting simulation run is broadened to a configuration consisting of round about 10 simulation runs.

REFERENCES

Akpan, J. I., and R. J. Brooks. 2005a. Practitioners' perception of the impacts of virtual reality on discrete-event simulation, In *Proc. of the 2005 Winter Simulation Conference*, 1976-1984.

Akpan, J. I., and R. J. Brooks. 2005b. Experimental investigation of the impacts of virtual reality on discrete-event simulation, In *Proc. of the 2005 Winter Simulation Conference*, 1968-1975.

Banks, J., J. S. Carson II, and B. L. Nelson. 1996. *Discrete-Event System Simulation*. Upper Saddle River, N.J.: Prentice Hall.

Barnes, M.R. 1997. An introduction to QUEST. In *Proc. of the 1997 Winter Simulation Conference*, 619--623.

Behzadan, A.H., and V. R. Kamat. 2005. Visualization of construction graphics in outdoor augmented reality. In *Proc. of the 2005 Winter Simulation Conference*, 1914 - 1920.

Chen, D., S. J. Turner, W. Cai, B. P. Gan, and M. Y. H. Low. 2005. Algorithms for HLA-based distributed simulation cloning. *ACM Transactions on Modeling and Computer Simulation* 15(4): 316 – 345.

Dangelmaier, W., D. Huber, C. Laroque, B. Mueck 2005. D³FACT insight – An immersive material flow simulator with multi-user support. In *Proc. of the 2005 Summer Computer Simulation Conference*, 239-242

Fischer, M., B. Mueck, K. Mahajan, M. Kortenjan, C. Laroque, and W. Dangelmaier. 2005. Multi-user support and motion planning of humans and humans driven vehicles in interactive 3D material flow simulations. In *Proc. of the 2005 Winter Simulation Conference*, 1921-1930.

Fujimoto, R. M. 1990. Parallel Discrete Event Simulation. *Communications of the ACM*, 33(10), 30-53.

Hlupic, V. 2000. Simulation software: An Operational Research Society survey of academic and industrial users. In *Proc. of the 2000 Winter SimulationConference,* 1676 – 1683.

Hybinette, M. 2004. Just-in-time cloning. In *Proc. of the 18 th Workshop on Parallel and Distributed Simulation,* 45–51.

Hybinette, M., and R. M. Fujimoto. 2001. Cloning parallel simulations. *ACM Transactions on Modeling and Computer Simulation* 11(4):378–407.

Kalyan S. P. 2006. Parallel and distributed simulation: traditional techniques and recent advances. In *Proc. of the 2006 Winter Simulation Conference*, 84-95.

Kamat, V. R., and J. C. Martinez. 2000. 3D Visualization of simulated construction operations, In *Proc. of the 2000 Winter Simulation Conference*, 1933-1937.

Schulze, T., S. Straßburger, and U. Klein. 1999. Online data processing in simulation models: New approaches and possibilities through HLA. In *Proc. of the 1999 Winter Simulation Conference*, 1602–1609.

Strassburger, S., T. Schulze, M. Lemessi, and G.D. Rehn. 2005. Temporally parallel coupling of discrete simulation systems with virtual reality systems. In *Proc. of the 2005 Winter Simulation Conference*, 1949-1957.

A Multi-Mode Mesh Generation Approach for Scientific Computing

F. Stimpfl, R. Heinzl, P. Schwaha, and S. Selberherr

Institute for Microelectronics, TU Wien
Gußhausstraße 27-29/E360, A-1040 Vienna, Austria
E-mail: {stimpfl|heinzl|schwaha|selberherr}@iue.tuwien.ac.at

KEYWORDS
Mesh generation, TCAD, Delaunay, Advancing front, Scientific Computing

ABSTRACT

With the steady evolution of software tools and programming languages, several programming paradigms have become available and new methodologies related to library centric application design have been developed. In addition to this paradigm shift, the increase of computing power requires an adaptation of well known techniques for parallel computing. Various problems of different disciplines appear, for which no efficient and definite solutions exist. This calls for new paradigms to be established and the development of new ways which fully exhaust the combination of sophisticated software and powerful hardware.

The area of spatial discretization, unstructured mesh generation, is one of these areas, which still exhibits various issues from fields such as modular software design, high performance, and most importantly robustness. In this work problems for mesh generation are identified and solutions explored, which anticipate the given challenges. This approach is combined in a software library based on a generic scientific simulation environment and compared to existing tools.

INTRODUCTION AND MOTIVATION

The field of scientific computing as a whole, and Technology Computer Aided Design (TCAD) in particular, relies on either domain-specific closed applications, or components tied very closely to specific representations which were often built for very fixed purposes and then wrapped into a large and monolithic toolkit. To this date, data structures and algorithms are implemented in a heavily application specific way, making their reuse practically impossible.

One of the most fundamental issues in TCAD, spatial discretization, requires certain and diverse constraints during mesh generation. Many software tools exist which solve very narrow sets of problems in special and well defined settings, but unfortunately most of the solutions to these problems are implemented repeatedly.

With the steady evolution of software engineering techniques and methodologies, new paradigms have emerged which offer great opportunities related to orthogonal module design and high performance approaches. Using a generic programming approach instead of object orientation, data structures and algorithms can be implemented orthogonally, where some languages, e.g. C++, offer additional high performance possibilities as well. In addition to these basic software engineering issues, the calculating power of modern computer systems, which has been doubling every 18 months in the past, and the emerging CPU parallelism, require different programming paradigms and application design methodologies.

These changes have to be taken into account as well as the upcoming shift to fully three-dimensional simulations [1]. This of course also implies, that three-dimensional mesh generation is getting more and more important in the next decade, especially in the field of TCAD where up to now, no comprehensive mesh generation application has emerged.

Mesh generation is a quite difficult task to be implemented on computers, because a rigoros algorithmic formulation does not exist. The emerging paradigm of computational topology will support mesh generation in the coming years [4, 17].

Different fields of TCAD application impose a variety of different constraints and requirements on mesh generation, e.g. topography simulation requires a good approximation of surface elements, while ion implantation simulation requires high density near the surface, according to the gradient of the ion distribution. Diffusion simulations add a need for a fine mesh at interfaces in addition to a high mesh density near the surface. The complex field of device modeling even requires a completely different type of mesh, necessitating a remeshing step for the whole input structure. In summary, it can be observed that each simulation step has completely different requirements on the underlying spatial discretization. Therefore, meshing is still one of the major showstoppers in the field of TCAD.

In this work we present a comprehensive modular, library centered approach which focuses on the problems described above. Already existing concepts and libraries in the field of scientific computing, which have proven to be successful, are combined on a new software basis.

PROBLEMS WITH MESH GENERATION

Before introducing different techniques of mesh generation, common problems are sketched which impose various constraints, restrictions, and difficulties for the task of mesh generation in general.

- **Different element size:** The generated mesh can consist of elements of different size, meaning, that adjacent elements vary greatly in size, e.g. a thin layer of oxide within a three-dimensional device, which is more than three order of magnitudes different in size.
- **Boundary requirements:** Various boundary requirements cause problems for mesh generation, e.g. a tetrahedra configuration known as Schönhart polyhedra [13] or the inclusion of preset surface edges, which imposes additional difficulties, if the surface requires Delaunay conformity [6].
- **Cospherical point sets:** Points are said to be cospherical, if at least n + 2 point are located on the perimeter of an n-dimensional sphere S, where S does not contain any other points of the mesh. This problem only arises, when the Delaunay criterion has to be met.
- **Degenerated cells:** Degenerated cells may result due to several reasons, such as the requirement to adhere to a given boundary or that the generated mesh has to include a given point cloud. The problem with these degenerated cells is that they may be unsuitable for the subsequent simulation tasks, e.g. interpolation of physical quantities. This problem increases further, as degenerated cells cannot be refined without adding additional degeneration. Therefore several classes of degeneration are classified and various mesh adaptation techniques are used to minimize the number of these elements.

As already pointed out, the application design for mesh generation not only has to deal with the given difficulties based on mesh generation itself, but it also requires various software engineering methodologies to develop robust applications.

MESHING METHODS

There are several methods for mesh generation, which are currently in use for scientific computing, such as:

- Giftwrapping
- Incremental Delaunay Triangulation
- Advancing Front
- Advancing Front with Delaunay Triangulation

To present the advantages and disadvantages of these different approaches, a brief overview of each is given in the following:

Giftwrapping

Starting with any element, a sphere is expanded around the element until the next point is found, where the extension of the sphere is controlled by a given direction vector. Figure 1 illustrates the idea behind the giftwrapping algorithm, which is suitable for use in multiple dimensions. Only convex hulls can be created using the giftwrapping algorithm and cospherical point sets impose additional difficulties.

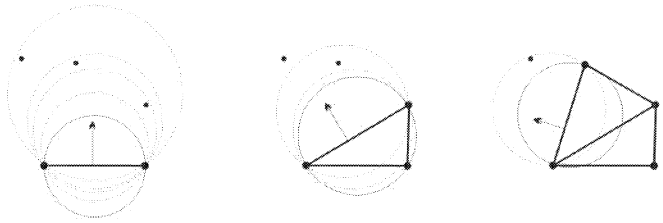


Figure 1: Giftwrapping example

Incremental Delaunay Triangulation

Many triangulations exist for a set of points in a plane. A triangulation which meets the so called circumcircle or Delaunay property is called a Delaunay triangulation. This property states that the circumcircles of the triangles may contain no points beside the points forming the triangles (see Figure 2).
One of the main advantages of the Delaunay triangulation in two dimensions is the minimization of the maximum angles and the maximization of the minimum angles. In three dimensions this advantages cannot be easily accomplished due to non-unique flip constraints [14].

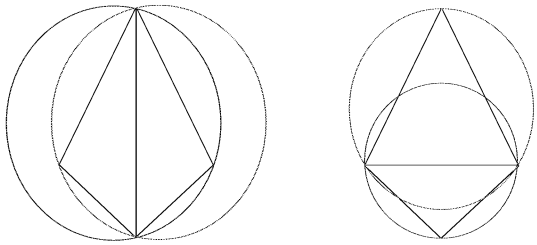


Figure 2: Delaunay property

An additional difficulty for this method is the incorporation of boundaries due to the fact that incremental Delaunay triangulation uses the convex hull of the input point set only. Given boundary representations have to be marked and reconstructed after the meshing process has completed.
Figure 2 shows two examples of a simple mesh. The left part violates the Delaunay criterion, because the cirumcircles include the points of both triangles. The right part is a correct Delaunay triangulation, because

the circumcircles of both triangles contain only their respective three points.

Advancing Front

The advancing front algorithm starts with a set of boundary elements of a given dimension, e.g. edges. These edges form the initial front which is advanced into the simulation domain. An edge of this set is chosen to form a new triangle, either with an existing point or a newly created point. The current edge is then removed from the front and the two new edges are, depending on their visibility, added to the front. This process terminates when no edges remain within the front.
Figure 3 shows a simple advancing front triangulation. The figure sketches the advancing front mesh generation concept in two dimensions. The dotted line represents the current front. New triangles are inserted, by joining the two ends of a front edge to either a newly created point or an existing point, one at a time [10].

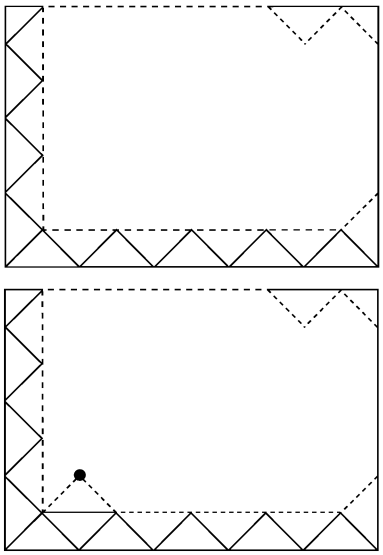


Figure 3: Advancing Front [10]

In contrast to the Delaunay triangulation, the advancing front method, easily incorporates the boundary. It starts from the given boundary representation and advances the front into the simulation domain. A major drawback of this method is that the quality of the generated elements heavily depends on the quality of the boundary elements. For different implementations of this type of mesh generation technique, the robustness issues are severe. The advantages of this method are the good control mechanism for the element sizes, the quality of the generated elements, and the not required Delaunay property which can optionally be incorporated easily. Parallelization can be quite challenging for surface creation, because new surface elements depend on the previously generated surface element. If the surface is partitioned, volume mesh generation, on the other hand, can be parallelized in order to distribute the generation process on several computers. Using the advancing front algorithm, it is no problem to divide the simulation domain into smaller domains and treat each domain on its own. However, boundary information, e.g. different materials, has to be available, otherwise the problem is similar as with surface mesh generation.
A method of incorporating the Delaunay property into the advancing front method has already been proposed [10], which adds new points ahead of the front and triangulates them according to the Delaunay criterion. This algorithm works like the conventional advancing front algorithm, only point insertion is extended to satisfy the Delaunay criterion.
Figure 4 shows the idea behind the algorithm. When inserting a new point, there are two possibilities:

- either the point is not inside any existing triangle circumcircle,
- or there exists at least one triangle, whose circumcircle contains this new point.

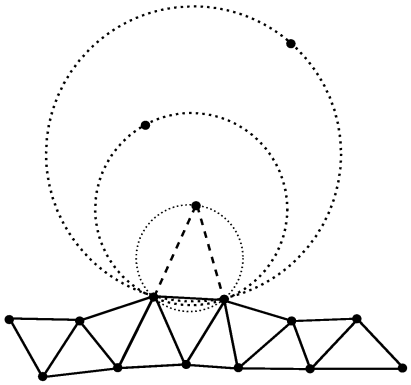


Figure 4: Advancing Front Delaunay Triangulation [10]

A different implementation is available using an abstract rule set [12], which describes different methods of controlling the advancing front. Basically, a rule describes where points are to be generated. The rule also manages changes in the advancing front. Some boundary elements have to be added while others have to be deleted. If no matching rule is found, the quality class is decreased. This allows the application of a rule of a lower quality level, which may also delete the current element.

SOFTWARE DESIGN

The meshing methods, described in the previous section, have been included as modules in the Generic Scientific Simulation Environment (GSSE [8]). The software design of our meshing library can be seen in Figure 5. The following presents our solution of the previously mentioned meshing problems.

- **Different element size**
  To cope with the problem of different element sizes we

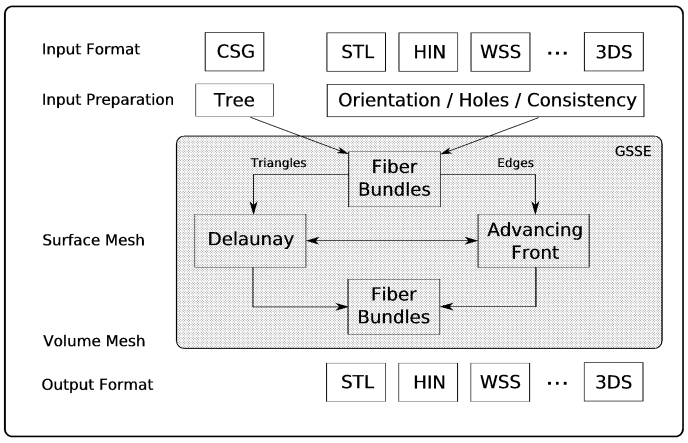


Figure 5: Design of GSSE

use the concept of a local feature size (lfs) [11]. The lfs of a point p is defined as the radius of the smallest disk centered at p, which intersects two non-incident vertices or segments of the planar straight line graph. The lfs guarantees that there are no abrupt changes in the size of the elements, which is especially important with the advancing front algorithm.

- **Boundary requirements**
  The application of the advancing front algorithm based on the introduced abstract rule mechanism easily incorporates the given boundary elements. Additionally, the Delaunay criterion can be used to create conforming boundary representations.
- **Cospherical point sets**
  This problem arises, when a mesh is used for, e.g. the finite volume method. The Voronoi graph [14] is therefore used to model flux conservation. Due to the fact that a Voronoi graph is the dual graph of a Delaunay graph, the input mesh has to exhibit the Delaunay property. Our approach optionally guarantees Delaunay conformity for both the surface and the volume.
- **Degenerated simplices**
  The use of the advancing front algorithm in combination with the already introduced quality rules and the corresponding point placement strategies reduces the degeneration of elements. The existing rules were extended to meet the Delaunay criterion.

Related to software engineering issues, the proposed solution uses a library centric application design paradigm [8] based on a multi-paradigm approach.

To incorporate various input formats and to solve the problems with different data models, the proposed approach uses a fiber bundle data model. Fiber bundles were first introduced by Butler and Pendley [3] and afterwards enhanced [2] to include cell complex properties. The basic approach of this data model is the separation of cells into a base space and the connectivity information storage within a fiber space.

All current data models can be mapped to the fiber bundle data model. This gives the advantage to read and write various file formats. Due to the modular design it is also possible to quickly write new input or output modules to accommodate an even wider range of file formats.

An advantage of our approach is, that the developed modules can be combined in various ways, depending on the requirements of the subsequent simulation. For example a mesh, initially generated using the advancing front algorithm, can be refined afterwards at certain points, which are interesting for the simulation, or the mesh can be modified to meet the Delaunay criterion. The module design can also be seen in Figure 5.

Another advantage of the modularized development approach is the possibility to interface different mesh engines and thereby fully utilize the orthogonal application approach.

Our library also contains a resampling step which consists of several substeps [12]. First the points are extracted from the input model. This extraction only applies to the 'points of intersection', which are the points where at least three surfaces intersect. To solve this problem, a bisection algorithm based on geometric tests is used. Afterwards, using this minimal mesh, a new surface is generated, with an advancing front algorithm [12].

MESHING LIBRARIES

As it is our goal to focus on library centric design, various meshing libraries are incorporated and therefore briefly presented here.

TetGen: is developed by Hang Si, from the Weierstrass Institute for Applied Analysis and Stochastics, and generates a Delaunay triangulation. Additionally, it supports the creation of Voronoi diagrams and convex hulls for three-dimensional point sets [5, 15, 16].

Netgen: uses the described abstract rules for advancing front mesh generation and offers a Delaunay point cloud generation module [12]. This module lacks the modeling of a guaranteed Delaunay volume mesh as well as the conforming Delaunay property for surfaces.

deLink: was developed for TCAD applications based on an advancing front method combined with a Delaunay method which does not create new points automatically [7]. Therefore, additional point clouds created separately have to be used to refine structures. The created meshes satisfy the Delaunay criterion for surfaces and volumes.

OUR APPROACH

It is important to highlight that TetGen and Netgen cannot deal with corrupted input representations, e.g. holes. deLink and GSSE use various methods to correct the input representations, where our approach can

also use a separate surface remeshing step. GSSE also incorporates constructive solid modeling [8]. Table 1 presents an overview of the features of each mesh generation module.

Features	TetGen	Netgen	deLink	GSSE
Delaunay	+	x	+	+
Advancing Front	-	+	-	+
Remesh Steps	-	+	-	+
Automatic Points	-	+	-	+
CSG Input	-	+	-	+
Polygon Input	-	-	+	+
Repair Surface	-	-	+	+
Multi Material	-	-	+	+

Table 1: Results of the comparison

EXAMPLES

This section presents some examples of meshes, created by the four previously outlined tools. The examples are analyzed and the quality distribution is presented. To compare the meshing tools, we categorize the elements of the mesh using two main parts [9]. First four classes of quality types, defined by the number of small dihedral angles, as shown in Figure 6 are set up.

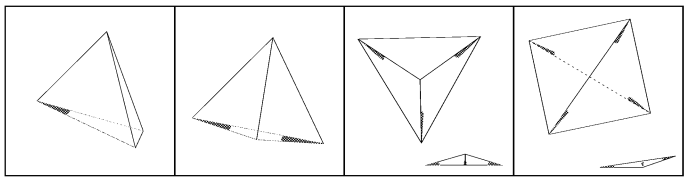


Figure 6: Four different classes of degenerated tetrahedra (wedge, spade, cap, sliver), sorted by the number of acute dihedral angles.

Then, the tetrahedra are classified by the number of degenerated triangles, like daggers and blades. The dagger has one short edge and at least one small angle, where the blade has no short edge and therefore one large and two small angles. Figure 7 shows the needle (or spire) with three daggers (the short edges are marked in the figure), the slat (or splinter) with two opposite short edges and, therefore, four daggers. Finally, the spindle does not have short edges and has therefore four blades as triangles.
The Figures 8 and 10 show the basic functionality and the boundary representation of the different meshing kernels. The Figures 9 and 11 summarize the respecitve statistics. The pictures in Figure 12 and 13 present examples from TCAD / topography simulation.

CONCLUSION

We have identified several problems of mesh generation and presented our solution to these problems. A meshing library based on GSSE has been developed to collect these solutions. Finally, a comparison between GSSE and established meshing kernels was performed showing, that GSSE has more mesh generation possibilities than the other investigated tools. The analysis of the examples also shows that GSSE produces an output of at least the same performance and even better quality than the other tools.

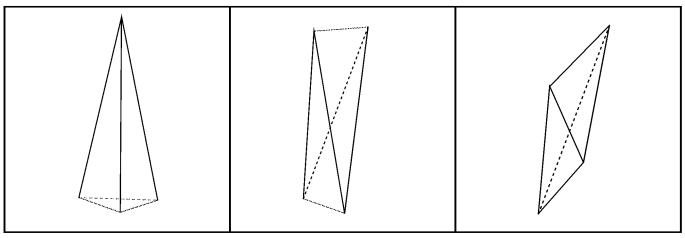


Figure 7: Three different types of degenerated tetrahedra (needle, slat, spindle).

Acknowledgment

This work has been supported by the Intel Corporation and the Austrian Science Fund FWF, project P19532-N13.

REFERENCES

[1] International Technology Roadmap for Semiconductors, 2005.

[2] W. Benger. *Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model*. Dissertation, Freie Universität Berlin, 2004.

[3] D. Butler and M. Pendley. A Visualization Model Based on the Mathematics of Fiber Bundles. *Computers in Physics*, 3(5):45–51, September/October 1989.

[4] H. Edelsbrunner and D. Guoy. An Experimental Study of Sliver Exudation, 2001.

[5] H. Edelsbrunner and N. R. Shah. Incremental Topological Flipping Works for Regular Triangulations. *Algorithmica*, (15):223–241, 1996.

[6] P. Fleischmann. *Mesh Generation for Technology CAD in Three Dimensions*. Dissertation, Technische Universität Wien, 2000.

[7] P. Fleischmann and S. Selberherr. Enhanced Advancing Front Delaunay Meshing in TCAD. In *Proc. Conf. on Sim. of Semiconductor Processes and Devices*, pages 99–102, 2002.

[8] R. Heinzl. *Concepts for Scientific Computing*. Dissertation, Technische Universität Wien, 2007.

[9] R. Heinzl and T. Grasser. Generalized Comprehensive Approach for Robust Three-Dimensional Mesh Generation for TCAD. In *Proc. Conf. on Sim. of Semiconductor Processes and Devices*, pages 211–214, Tokio, September 2005.

[10] D. Mavriplis. An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness. *J. Comput. Phys.*, 117(1):90–101, 1995.

[11] J. Ruppert. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *J. of Algorithms*, 18(3):548–585, 1995.

[12] J. Schöberl. NETGEN - An Advancing Front 2D/3D-Mesh Generator Based on Abstract Rules. *Comput. Visual. Sci.*, 1:41–52, 1997.

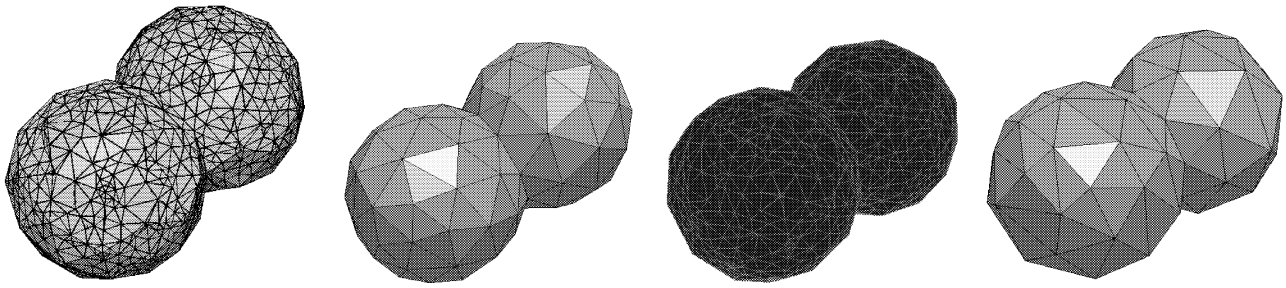


Figure 8: A simple sphere, but yet a complex task for a mesh generator, is used to highlight the different approaches related to the investigated types of meshing algorithms, from left to right: TetGen, Netgen, deLink, GSSE.

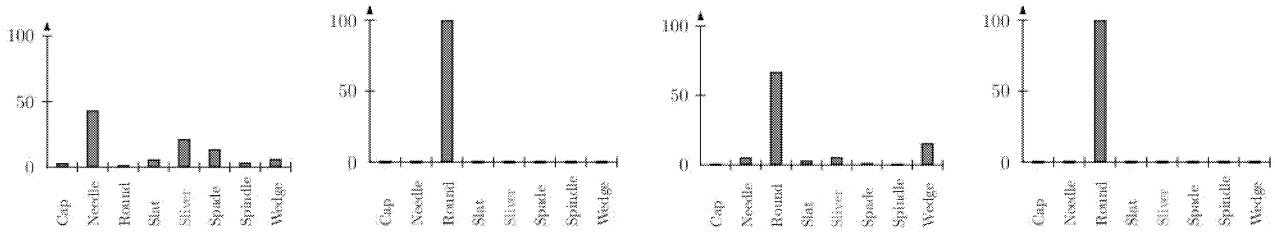


Figure 9: Statistics for the meshing example in Figure 8, from left to right: TetGen, Netgen, deLink, GSSE.

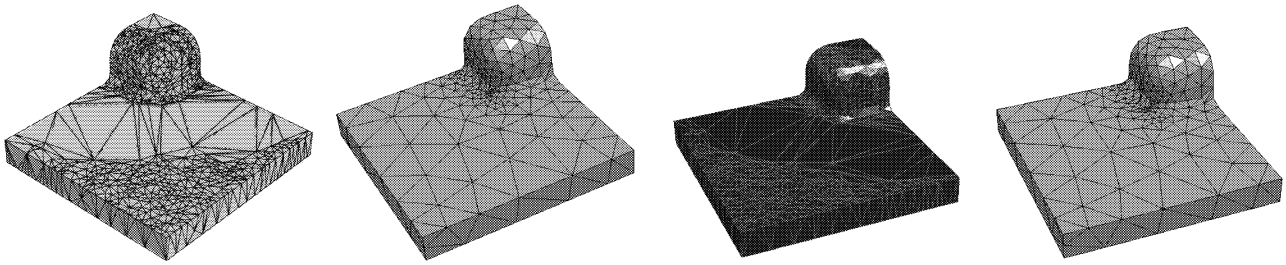


Figure 10: An example from topography simulation is used as example to illustrate the different types of meshing results, from left to right: TetGen, Netgen, deLink, GSSE.

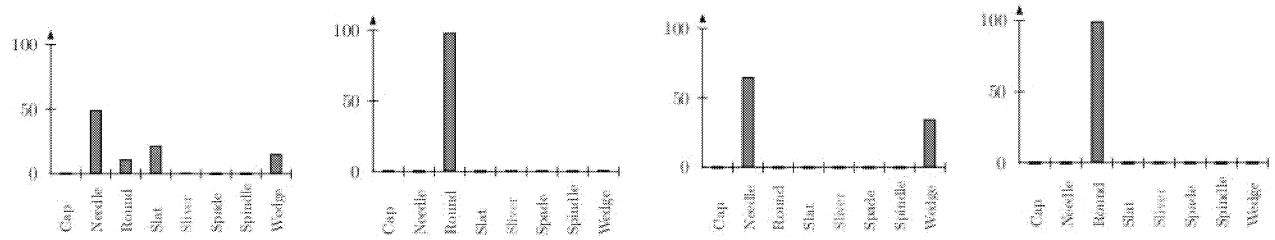


Figure 11: Statistics for the meshing example in Figure 10, from left to right: TetGen, Netgen, deLink, GSSE.

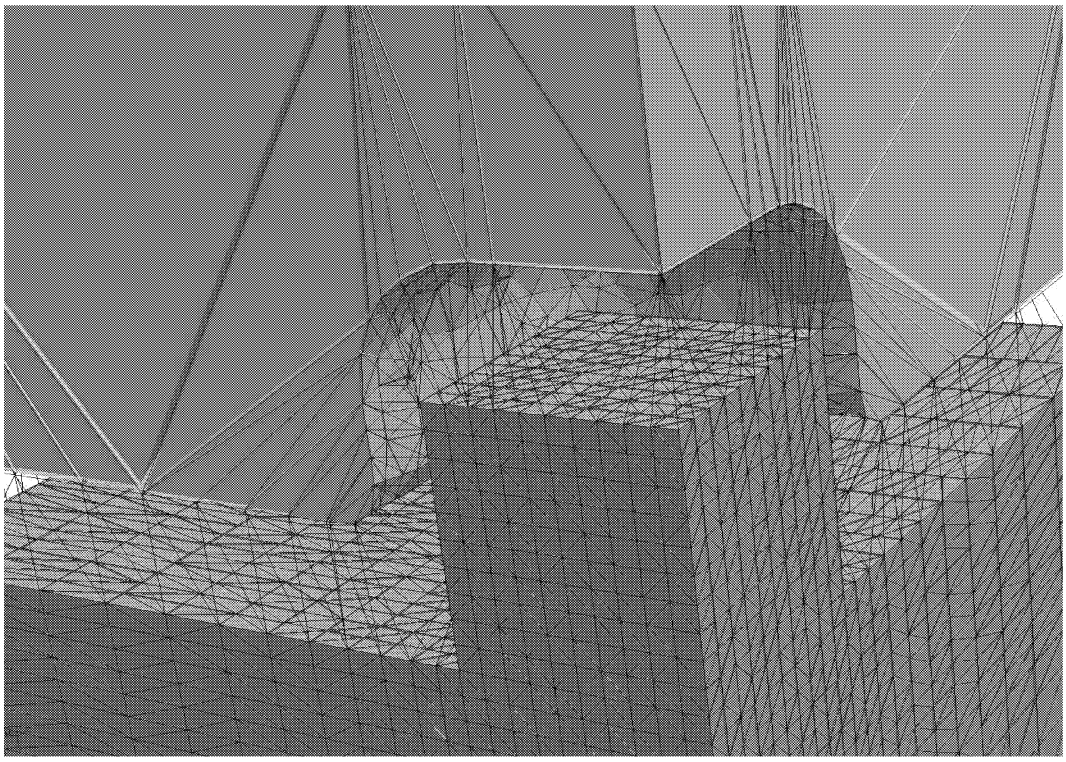


Figure 12: The complete example from topography simulation with the additional constraint of including a given point cloud as can be seen in the bottom part of the structure. The middle part is made transparent to expose the surface mesh. The top part is not required for subsequent simulation and is therefore meshed without quality constraints.

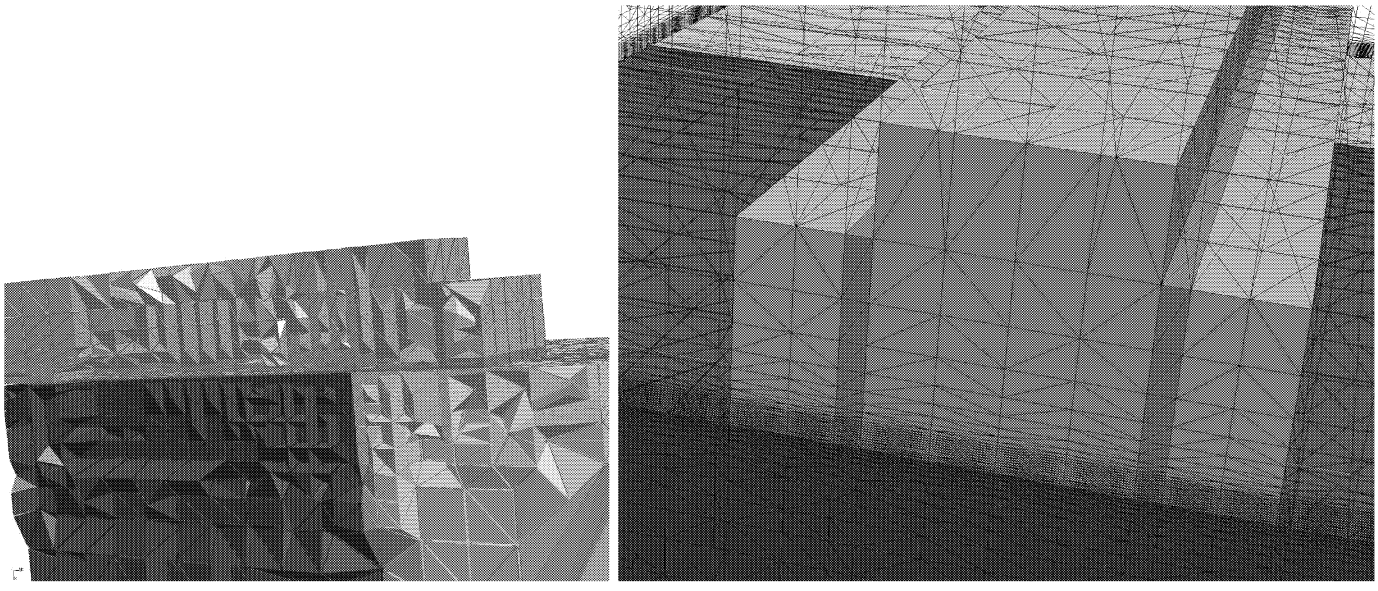


Figure 13: A three-dimensional device structure for a MOSFET with an additional externally supplied point cloud. The important part is the regularity of the elements in the channel region. The different aspect ratios, e.g. the thin red oxide part and the large blue silicon part, are also an additional complication for the mesh generation algorithm.

[13] E. Schönhart. Über die Zerlegung von Dreieckspolyedern in Tetraeder. In *Journal Mathematische Annalen*, March 1928.

[14] J. R. Shewchuk. *Delaunay Refinement Mesh Generation.* Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997.

[15] H. Si. On Refinement of Constrained Delaunay Tetrahedralizations. *15th Intl. Meshing Roundtable*, September 2006.

[16] H. Si and K. Gaertner. Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. *14th Intl. Meshing Roundtable*, pages 147–163, September 2005.

[17] A. J. Zomorodian. Topology for Computing. In *Cambridge Monographs on Applied and Computational Mathematics*, 2005.

BIOGRAPHIES

FRANZ STIMPFL was born in Vienna, Austria, in 1980. He studied computer science at the Technical University Vienna. He joined the Institut für Mikroelektronik in September 2007, where he is currently working on his doctoral degree.

RENÉ HEINZL studied electrical engineering at the Technical University Vienna. He joined the Institute for Microelectronics in November 2003, where he finished his doctoral degree in September 2007. In April 2005 he achieved first place at the doctoral competition at the EEICT in Brno. His research interests include computational science, programming paradigms, high performance programming techniques, solid modeling, scientific visualization, and algebraic topology for TCAD.

PHILIPP SCHWAHA studied electrical engineering at the Technical University Vienna. He joined the Institute for Microelectronics in June 2004, where he is currently working on his doctoral degree. His research activities include circuit and device simulation, device modeling, and software development.

SIEGFRIED SELBERHERR received the Ph.D. degree in technical sciences from the Technical University Vienna in 1981. Since that time he has been with the Technical University Vienna as professor. Dr. Selberherr has been holding the "venia docendi" on "Computer-Aided Design" since 1984. As of 1988 he has been chair professor of the Institut für Mikroelektronik. From 1998 to 2005 he served as Dean of the "Fakultät für Elektrotechnik und Informationstechnik" at the Technical University Vienna. His current topics of interest are modeling and simulation of problems for microelectronics engineering.

Modular L-Systems: Generating Procedural Models using an Integrated Approach

J. L. Hidalgo and E. Camahort and F. Abad and R. Vivó
Dpto. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, 46021 Valencia, Spain

KEYWORDS
L-systems, Procedural Models, Simulation for Graphics

ABSTRACT

A modeling technique commonly used in Computer Graphics and Simulation is L-systems. L-systems are procedural generators that fall into the category of growth simulators. They can generate complex geometries like plants and trees, shells and coral reefs, cities and landscapes, and even texture images. L-systems are typically implemented as a set of ad-hoc tools specific to each application area. Users are restricted to the language provided by the system and must follow inflexible guidelines to add their own code to generate new models. In many cases this is not even possible.

Our approach to L-system based modeling aims to bridge the world of rewriting engines and the world of imperative programming using Object-Oriented languages. We build L-systems using a modular approach that simplifies and speeds up construction of complex geometric models. We call it Modular L-systems (ML-systems) and support not only construction, but also integration of different types of models into one scene. A system's rules are written in a scripting language that is independent from the objects handled. The objects are encoded in C/C++ methods and classes, thus supporting the expressiveness of any high-level L-system. We demonstrate how to model objects with ML-systems by simulating the growth of crystals, coral reefs and other geometries using the same tool.

Introduction

Many simulation application areas require generating automatic content. Movies, console and computer games, and massive on-line games also require large numbers of object models and complex simulations. Models may contain millions of instances of an object, like crowd models made of people, and city models made of streets and buildings.

People, for example, can be represented using the same geometry. Each each person, however, has slightly different properties, appearances and behaviours. Modeling such objects by hand is impractical due to their complexity and large number of instances. We need systems to generate models automatically.

Early automatic model generation was done using procedural techniques like fractals [8], particle systems [13], and grammar-based systems [12]. These can generate texture images, terrain models, water and rain, hair, plants and trees, and urban landscapes. Early models were generated using a grammar-based technique called L-systems [6].

L-systems are made of an axiom and a set of production rules. Rules are applied to the axiom to obtain a derived string which is interpreted to produce a geometric or image-based model. L-systems can be parameterized to generate different instances of the same object type; they are called parametric L-systems.

Geometry generators based on L-systems are usually targeted at specific applications with fixed symbols and tools based on LOGO's turtle metaphor. They are highly dependable on the rewriting engine, thus preventing grammar improvement, language extensions, and code reuse. These generators can not generate different models for the same application. They require multiple L-systems that are difficult to integrate within the same application framework.

To overcome these problems we present modular L-systems (ML-systems), a new procedural technique that builds general L-systems easily extensible and suitable for different applications. ML-systems can generate different types of models using the same rewriting engine. This engine is written in C/C++ and uses Lua [3] as scripting language.

ML-system rewriting and interpretation rules access C/C++ user objects using bindings. Bindings support dynamic loading and generation of different object types using a single tool. For example, we may model an urban environment with different types of objects like streets, cars, trees, houses and buildings, each type generated using a different L-system. We may also generate procedural textures and metaball-based coral reefs using the same formalism and tools. Finally, we may place meshes within landscapes, simulate growth of organic structures, and automatic generate game levels [14].

Our tool is designed with reusability in mind: objects, rules and bindings are organized in plugins that may be loaded for different applications. Object modeling is decoupled from string rewriting and offers a flexibility unavailable in other systems. Our system is described in this paper, starting with the definition of ML-objects and rules, continuing with rewriting and interpretation,

and concluding with several application examples. The paper also reviews previous work in the area and proposes directions for future work.

Previous Work

L-systems are a modeling technique that runs programs to generate graphical objects like images, geometry and particles that produce, for example, a plant ecosystem [12]. It requires little storage and supports dynamic growth, thus producing very compact models. An early example of procedural models can be found in [13].
L-systems are procedural models based on grammars. Made of an axiom and rewriting rules, they generate objects by deriving and interpreting grammar strings. L-systems were introduced by Lindenmayer to model cellular interaction [6]. They have been applied to plant and tree modeling [11], shell texturing [2], virtual urban landscaping [10], and geometry mesh generation [15].
L-systems were later parameterized to improve their expressiveness, allowing arithmetic and boolean expressions on the parameters during the rewriting process [1] [7]. Stochastic L-systems were also proposed that replaced parameters by random variables [4]. These systems are all based on the turtle metaphor [11].
More recently the L+C language was proposed [5]. In L+C symbols are C data structures, and L-systems are translated into C code. A rule's right-hand side is also made of C code from user developed C libraries. This improves on earlier systems by adding computation and data management to the rewriting process. The L+C language, however, does not benefit from the advantages of scripting languages since it requires models to be generated in C, then compiled and added to a library that is linked to the main system.
Finally, the FL-System was introduced that departed from the turtle metaphor [9]. It interprets the symbols of the derived string as function calls that can generate any geometry type. The authors use it to generate VRML buildings. The system has a limited expressiveness because it generates VRML nodes but does not allow adding functionality outside of VRML.

Modular L-Systems

We want the advantages of FL-systems and L+C systems without their drawbacks. Like an FL-system, we want to generate geometry without the turtle metaphor. But we also want a system with a complete and extensible rewriting engine.
As the L+C system, we want output code with arithmetic and boolean expressions, flow control statements and other imperative language structures. But we do not want to compile and link to develop and test every object. Instead, we use a scripting language for fast prototyping and we use plugins and C/C++ bindings to handle objects that are instances of OOP classes. This gives us as much expressiveness as the L+C system without the limitations of its strict grammar.
So, we introduce Modular L-systems, a class of parametric L-systems whose rewriting rules control user objects implemented in an imperative language. These objects are called ML-objects and include: C/C++ code, events, the rewriting string, and rewriting and interpretation rules. Rules are written in the Lua [3] scripting language. Thus, rules can be changed on the fly because recompiling is unnecessary. Lua rules can access the user code written in C/C++ through a set of bindings.
The advantage of ML-Systems is the use of the scripting language to handle any user object. It acts as a middleware layer providing a uniform access to the C/C++ objects. This improves on other system's flexibility and expressiveness because it supports loading dynamic objects, running code to create instances, calling object methods, and accessing the objects' state. We can handle any object and its methods using the same axiom, rules, rewriting engine and interpretation. The only difference is how the user implements the C/C++ objects. The user objects are loaded during the initialization of the ML-system. They are contained in plugins that include the classes, its bindings and wrappers for the user's C/C++ code.
With this approach we offer a platform that handles user code instead of a fixed set of tools targeted at a specific application. By using general C/C++, code an ML-system can be adapted to any application domain. It can handle multiple domains at once, and it can target problem's solution to different application domains. Different object types can coexist in an ML-system. This is why ML-systems are more flexible and more expressive than previous approaches.

Defining ML-Objects and Rules

Each ML-object is a Lua script with different parts. It starts with an initialization where the user loads libraries, plugins and other Lua modules. It also sets initial values and declares grammar symbols. The ML-object then defines three functions: `createInstance` creates a new instance of the ML-object, `beforeStep` is a function run before each derivation/interpretation step, and `afterStep` is a function run after each step.
`CreateInstance` is the only required function. It returns the instance, its global environment and the default axiom. The `beforeStep` function is responsible for resetting the ML-object to its initial state. This happens before each derivation and interpretation step. It implies that, in most cases, the object needs to be re-created.
Two more functions create rewriting and interpretation rules. Both receive the left hand side (LHS) and the right hand side (RHS) of the rule. The LHS is an string that defines the rewriting symbol and its contexts. The symbols of a string have associated Lua dictionaries con-

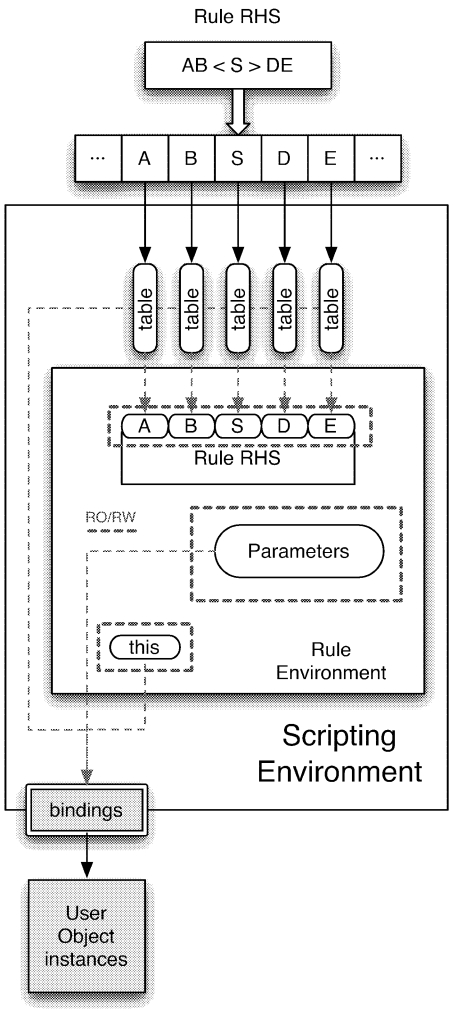


Figure 1: Rule matching mechanism. State surrounded by dashed boxes is read-only for rewriting rules and read-write for interpretation rules.

taining sets of labeled items. The RHS is the Lua function executed when the LHS matches.

Additionally, a global environment table holds global variables that can be used during the derivation/interpretation process. Its contents enables the parametrization of the derivation process.

Rewriting and Interpretation

ML-System rules are chunks of Lua code that can access C/C++ objects through the bindings. Once we have loaded the plugins, created the objects and declared its axiom, rules are executed to modify the derivation string and do the interpretation. Each derivation step rewrites and interprets all the symbols of a given rewriting string.

ML-System uses two types of rules. Rewriting rules modify the derivation string without changing the C/C++ objects' state and the Lua tables. Interpretation rules change the objects' state without modifying the derivation string.

Both types of rules have a left-hand side (LHS) and a right-hand side (RHS). The LHS has the form "AB < S > DE" where S is the symbol being rewritten and AB and DE are the left and right contexts, respectively. These contexts are optional. To match a rule we compare the rewriting string with the LHS's symbol and its contexts.

The RHS of the rules is a Lua function that computes the result of the rewriting process or does the interpretation. These functions can access C/C++ objects through the bindings. They can also access the global environment that we created during the object's instantiation.

The RHS of the rules can also access the LHS symbol's tables and the object being derived through the variable *this*. Figure 1 shows the rule matching mechanism.

When a rule matches, the system places in the rule's scope variables with the same name as the symbols of the LHS. These variables are always in read-only mode so their values can not be modified. The object associated to the rule (*this*) is also placed in the rule's scope as well as the rule's RHS parameters. If the rule is a derivation rule, the object will be a *const* reference so only *const* declared methods could be used. If the rule is an interpretation rule the object will be fully accessible.

The matching mechanism only compares the symbols with the LHS of the rule. Depending on the current symbol several rules may be applicable. The first rule that does not return *false* is applied. Rules are traversed in the same order they were declared in the object's definition script.

Rewriting Rules

Rewriting rules take the derivation string and replace one symbol with a new set of symbols or the empty string. They do not modify the rest of the state of its associated ML-object. For example, the following Lua code describes two rewriting rules, one for symbol `A` and one for symbol `B`:

```
RRule("A",
   function()
     return {B{},T{v=1},A{},T{v=1},B{}}
   end
)
RRule("B",
   function()
     return {A{},T{v=-1},B{},T{v=-1},A{}}
   end
)
```

Note that the RHSs are functions that return new symbols that may be parameterized. The parameters are pairs (key = value) contained in the entries of the symbol's dictionary table. Both keys and values can be of any valid Lua type, that is numbers, strings, other tables, functions, or user objects. To erase the symbol

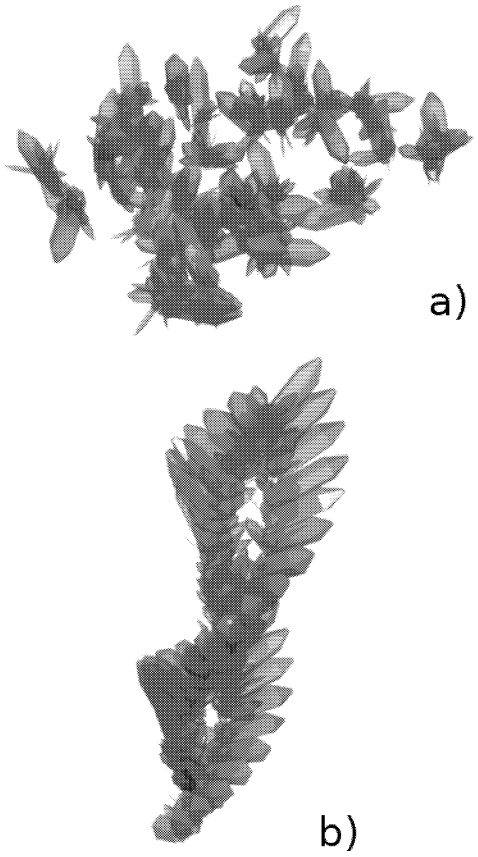


Figure 2: Crystals with the extruder plugin. a) some crystals spread around, b) crystals forming a tower.

from the current derivation string a function should return an empty table {}.

Interpretation Rules

Interpretation rules are executed after each rewriting step. These rules do not modify the derivation string, they only affect the state of the C/C++ objects. These rules return *false*/*nil* to indicate that they can not be applied to the current symbol. Otherwise, it is assumed they can be applied.
Interpretation rules are allowed to change the state of the user object through the reference "*this*", this reference is the same value the function beforeStep or createInstance returned. These rules are applied sequentially in the order of the symbols of the derivation string.

Derivation and Interpretation

Using a combination of rewriting rules and interpretation rules, ML-Systems can handle user's objects and modify them. The derivation and interpretation process is executed interlaced in each step.
First we call the `beforeStep` function of the object. This usually resets the object to its initial state. Then

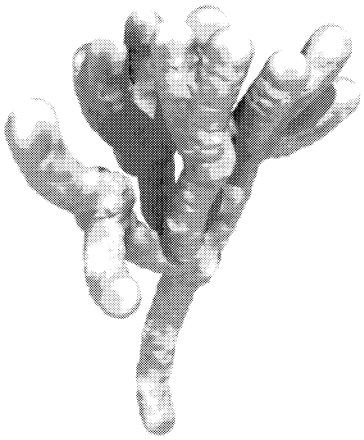


Figure 3: Example of a Metaball object, part of a coral reef.

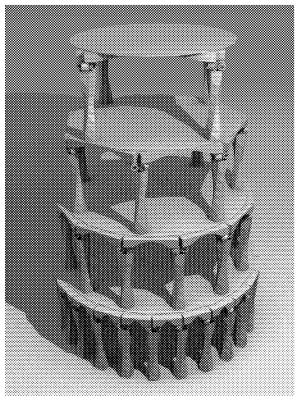


Figure 4: A tower generated using the extruder object.

we enter a loop where we rewrite one symbol and then interpret the same symbol using a rule selected using the matching mechanism explained earlier in this Section.
After the derivation of the last symbol we have in the new derivation string the result of the whole process. Finally the function `afterStep` is called, completing a full derivation/interpretation step.

Application Examples

We introduce two examples of C++ objects used to generate geometric models for the images of this paper. The objects are an extruder and a metaball. The Extruder plugin creates 3D meshes by extruding shapes on a 2D plane. It has methods to manipulate the extruded shape, each of its points, the matrices that controls the shape ortientation and the direction and length of the extrusion.
The Metaball plugin generates particles with positive and negative charges. Then it positions them in 3D space. Given a specific charge, the plugin generates the

isosurface containing the points with that charge in the space. The extruder and metaball objects are unrelated to L-systems. They belong to C++ classes that generate 3D geometry based on user defined ML-objects. For example, Figures 2 to 4 show three examples geometries generated with these objects.

Conclusions and Future Work

We have introduced Modular L-Systems, a generalization of L-Systems that can be easily extended to be used for different applications. ML-systems can generate different types of complex models using a single rewriting engine written in Lua, a scripting language. We provide bindings to access C/C++ user code within the rewriting engine. This allows different types of objects to be dynamically loaded. These objects can be implemented using object-oriented programming languages.
Our approach bridges the gap between rewriting engines and object models programmed in imperative languages. For example, we can use Modular L-systems to model terrain, plants and trees, coral reefs, shells, textures and other models. We have demostrated the application of ML-Systems to the generation of three types of geometric objects.
We plan to use ML-systems to generate hierarchical models, like a city made of buildings, which in turn are made of floors, which in turn are made of rooms, etc. We can also design the wallpaper of the rooms using image generating ML-systems. Finally, we can generate the behavior of the city's inhabitants using an ML-system. All of these may be done dynamically or on demand.

Acknowledgements

This work was partially supported by grant TIN2005-08863-C03-01 of the Spanish Ministry of Education and Science, and the GAMETOOLS STREP project IST-004363 of the 6th Framework Program of the EU.

REFERENCES

[1] R. Baker and G. T. Herman. Simulation of organisms using a developmental model, parts I and II. *International Journal of Bio-Medical Computing*, pages 201–215 and 251–267, 1972.

[2] Deborah R. Fowler, Hans Meinhardt, and Przemyslaw Prusinkiewicz. Modeling seashells. *Computer Graphics*, 26(2):379–387, 1992.

[3] Roberto Ierusalimschy. *Programming in Lua 2ed.* Lua.org, 2006.

[4] H. Jones and D. Saupe. Stochastic methods and natural phenomena. In *Tutorial Note 2. Eurographics'91*, 1991.

[5] R. Karwowski and P. Prusinkiewicz. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science*, 86(2):141–159, 2003.

[6] A. Lindenmayer. Mathematical models for cellular interaction in development, parts I and II. *Journal of Theoretical Biology*, (18):280–315, 1968.

[7] A. Lindenmayer. Adding continuous components to L-systems. In G. Rozenberg and A. Salomaa, editors, *L Systems*, Lecture Notes in Computer Science 15, pages 53–68. Springer-Verlag, Berlin, 1974.

[8] B. B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman and Company, 1982.

[9] Jean-Eudes Marvie, Julien Perret, and Kadi Bouatouch. The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer*, 21(5):329–339, 2005.

[10] Yoav I. H. Parish and Pascal Müller. Procedural modeling of cities. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308, New York, NY, USA, 2001. ACM Press.

[11] P. Prusinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 – Vision Interface '86*, 1986.

[12] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants.* Springer-Verlag, New York, 1990.

[13] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 313–322, New York, NY, USA, 1985. ACM Press.

[14] T. Roden and I. Parberry. *Game Programming Gems 5*, chapter 5.9 Procedural Level Generation. Charles River Media, 2005.

[15] Robert F. Tobler, Stefan Maierhofer, and Alexander Wilkie. Mesh-based parametrized L-systems and generalized subdivision for generating complex geometry. *International Journal of Shape Modeling*, 8(2), 2002.

A MARKET MODEL FOR THE DIDACTIC INDUSTRIAL MANAGEMENT SIMULATION

Peter Lawrence
School of Business and Informatics
Australian Catholic University, St. Patrick Campus
Fitzroy, Victoria 3065 AUSTRALIA
Email: peter.lawrence@acu.edu.au

KEYWORDS

Education, Simulators, Management Games, Business, Model Design.

ABSTRACT

The Didactic Industrial Management Simulation (DIMSIM) was developed as an experiential learning case study for a course in discrete event simulation. It allows students to conduct their practical simulation project in the context of a more realistic business environment than allowed by more traditional, paper-based case study approaches. A major part of this realism derives from students' need to operate in a competitive market environment which responds dynamically to the business decisions made by participants in the exercise. To achieve this, a mathematical market demand model was empirically developed and incorporated into DIMSIM. This paper describes the structure and operation of the various components of that model and discusses some of its shortcomings and potential future improvements.

BUSINESS SIMULATIONS

The use of business simulation games in management education is well documented (Burgess 1991; Faria 1987, 1989; McKenna 1991) and dates back at least to the development of early commercial computer systems in the 1950s (Lawrence 1997). Numerous examples have been developed and used around the world since then and many have enjoyed success as commercial publications such as "The Business Policy Game" (Cotter and Fritzsche 1995). Typically, these simulation games require participants to make a variety of regular strategic management decisions such as setting product prices, organising operational financing, production scheduling and human resource planning and policies. Similarly, simulation exercises have been used as aids in teaching various quantitative analytical techniques in operations management such as production scheduling (Chu, Huttenstein and Greenlaw 1996). Students see simulations as enjoyable and effective learning devices (Ledman 2005) and studies have suggested that they are more effective than paper-based case studies at developing conceptual knowledge (Wolfe 1975; Keys and Bell 1977).

THE SIMULATION

The Didactic Industrial Management Simulation (DIMSIM) was developed by the author to combine the advantages of both strategic and operational management simulations and allow students to see directly the strategic business implications of their operational modelling (Lawrence 2002) and hence complement a course in discrete event simulation where, as a major project, students were required to perform a complete stochastic simulation study of DIMSIM's manufacturing production environment.

DIMSIM alleviates many of the shortcomings of traditional simulation exercises by putting the project into a more realistic context. Students are faced not merely with a static description of a production environment but a complex dynamic model of a competitive business environment that includes not just their organisation's production facilities but also its financial, marketing and human resource management functions. Students, or student teams, commence the exercise as executive managers in one of up to six manufacturing/marketing enterprises engaged in an oligopoly.

Initially, all companies are identical in terms of their inventory, assets, wealth and market share. Students then make a series of quarterly decisions that guide each company on its independent path for the next three years of trading.

Figure 1 shows the dialogue through which students enter their business decisions. As can be seen, in addition to production and related HR decisions, appropriate values must be set for marketing variables such as product price, sales representatives' salaries and commissions and financing variables such as investment decisions and share and bond issues.

When all students have submitted their decisions for the upcoming quarter DIMSIM processes them and determines the state of each company at quarter's end. This involves not merely calculating the company's production output but also determining, through a competitive market model influenced by various economic variables such as inflation and GDP growth, the relative sales performance of each company in the marketplace and hence it's new financial position.

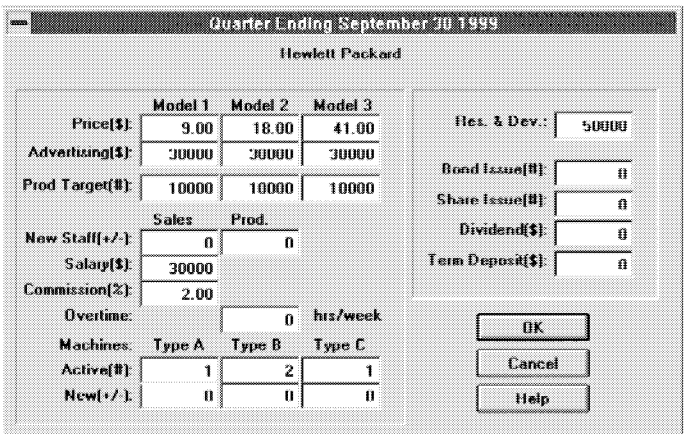


Figure 1 The DIMSIM Decision Entry Dialogue

A key part of the realism of the exercise is in giving students the impression that they are competing head to head in a real market. To achieve this, DIMSIM needs at its heart a mathematical model of a closed competitive market that calculates each company's product demand as a function of the decisions made, not just by that company, but by all participants in the simulation along with various environmental factors. What follows is a description of the market model that was developed to satisfy that requirement.

THE MARKET SHARE MODEL

The market consists of up to 6 companies each selling up to three models of product: 1-economy, 2-quality and 3-luxury. The input variables which determine sales for the quarter are:

Product Price
Each company sets a selling price for each of the models it produces. These prices will generally start at around:

$15.00 for model 1,
$30.00 for model 2,
$75.00 for model 3.

Advertising
Each company sets an advertising budget for each model which it produces. The effect of the advertising dollar drops off at around $100,000.

No. of Sales Representatives.
This is the number of sales reps working for each company. There are no limits but if there are no reps then no sales can occur.

Sales Representatives' Base Salary
The more reps are paid as a base salary the less incentive they have to sell and so sales drop. However, if a rep's total package is too low by industry standards then he or she is likely to resign thus increasing training and hiring costs. This figure will normally be between $20,000 and $60,000.

Sales Representatives' Commission
Reps are paid a percentage of the revenue they generate. This can be anywhere from 1% to 10%.

As the simulation progresses, all monetary amounts are corrected for inflation by dividing by the CPI which is unity at the beginning.

Calculation of the final quarterly sales figures is carried out in a four step process:

1. Calculate the total market size, that is, the total units sold across all companies and models.
2. Determine the relative sales for each of the three models across the total market.
3. Apportion the sales of each model between the companies in the market.
4. Determine if any company has unsatisfied demand and, if so, distribute that demand to companies who can meet it.

Each of these four steps is detailed in the following sections.

Step1: Market Size
The total market size, the sum of all sales of all models, is controlled by industry wide aggregates and averages of the decision variables as follows.

The average price of product in quarter k is,

$$\overline{P_k} = \frac{\sum_{i=1,j=1}^{n,m} p_{ijk}}{nmC_k}$$

where p_{ijk} is the price, in dollars, of model j from company i, in quarter k (assuming m models and n companies)[1] and C_k is the CPI in quarter k relative to the start of the simulation.

The total, industry wide, advertising expenditure in any given quarter is,

$$A_k = \frac{\sum_{i=1,j=1}^{n,m} a_{ijk}}{C_k}$$

where a_{ijk} is the advertising budget of company i for model j in quarter k.. However, advertising effect does not end abruptly at the end of the quarter in which it is spent but will linger and diminish as time passes. To simulate this effect it is calculated as a weighted moving average over a whole year. Therefore, the total industry advertising effect for quarter k is given by,

[1] If any company does not produce a particular model then that price is excluded from the calculation.

$$A_k = \frac{\sum_{i=1,j=1}^{n,m} \bar{a}_{ijk}}{C_k}$$

where,

$$\bar{a}_{ijk} = 0.4a_{ijk} + 0.3a_{ijk-1} + 0.2a_{ijk-2} + 0.1a_{ijk-3}$$

The total number of sales reps in the field during quarter *k* is,

$$R_k = \sum_{i=1}^{n} r_{ik}$$

where r_{ik} is the number of reps employed by company *i* during quarter *k*. All reps sell all models of product, assuming their company produces it.

The total market demand (number of units sold if stock is available) is determined by the product of three factors: the price factor F_p, the advertising factor F_a and the sales reps factor F_r. In addition, market size is compounded by economic growth and so the total market demand, *D*, for any given quarter[2] is the product of four factors,

$$D = F_p F_a F_r F_e$$

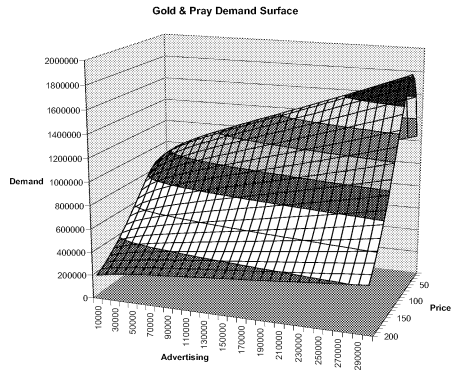


Figure 2 Gold & Pray Demand Surface Showing Breakdown of the Relationship at Low Price Levels

A common approach to this market size calculation is that of Gold and Pray (1983) who used power law equations for the factors in their frequently cited demand equation,

$$Q(P,M,R) = g_1 P^{-(g_2+g_3 P)} M^{(g_4+g_5 M)} R^{(g_6+g_7 R)}$$

relating total industry demand (Q) to price (P), marketing (M) and research (R). However, as has been observed by numerous authors (Perotti and Pray 2000; Murff, Teach and Schwartz 2006), this model can become unstable if decision variables are extrapolated beyond their intended range. This can be seen at low price levels in Figure 2.

To avoid this problem, the factor definitions used in DIMSIM were designed to be monotonic, with the exception of the advertising factor as explained below. The factors were thus defined as follows.

Market Size Price Factor

$$F_p = 1 + 2e^{-0.017\bar{P}}$$

Market Size Advertising Factor

$$F_a = 7.5\times10^5\left[\tanh\left(\frac{A}{7.5\times10^5}\right) - 0.1\left(1+\tanh\left(\frac{A}{7.5\times10^5} - 2.0\right)\right)\right]$$

Market Size Sales Representative Factor

$$F_r = 1 - e^{-0.02R}$$

Market Size Economic Growth Factor

$$F_e = \eta\prod_{l=1}^{k}(1+g_l)$$

This final factor compounds economic growth. GDP growth for the l^{th} quarter is represented by g_l where l represents the set of all quarters from the commencement of the simulation up to and including the current quarter, *k*. The term η accounts for the fact that the market for this type of product may grow or shrink at a different rate than the economy as a whole. If $\eta > 1$ then demand grows/shrinks faster than the economy.

CPI, growth and the factor η are variables assigned by the simulation administrator. CPI and growth are reported to participants at the end of each quarter but η is treated as a purely internal parameter the value of which is not directly divulged.

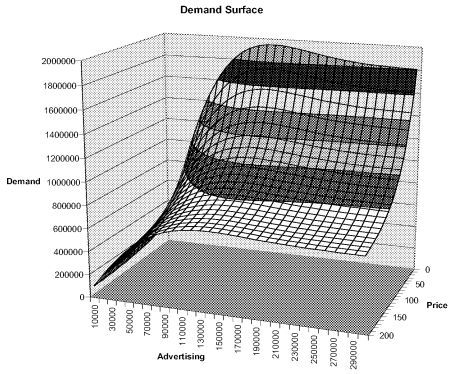


Figure 3 DIMSIM Demand Surface Response to Price and Advertising.

[2] For simplicity subsequent equations exclude the *k* subscript identifying the quarter unless necessary.

While the form of the advertising factor equation may seem complex it allows realistic modelling of the effect of advertising on demand. As expenditure increases so does demand, but with a diminishing rate of return. Eventually, excessive advertising can lead to consumer fatigue (Kamp and Kaiser 1999) and this compound sigmoidal form of the demand curve allows that fatigue to produce negative demand elasticity, as can be seen in Figure 4.

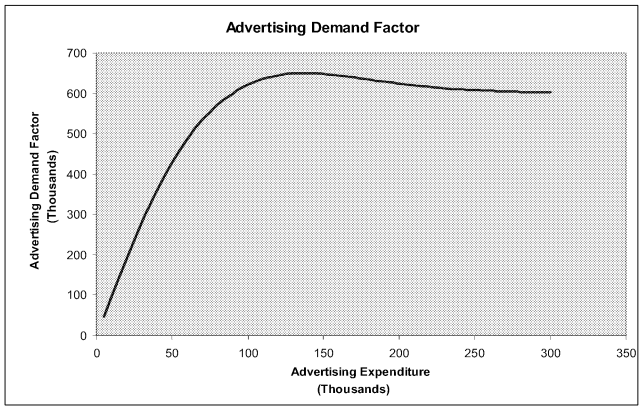


Figure 4 Effect of Advertising Expenditure on Demand.

Step 2: Relative Market Shares of Each Model

Having determined the total size of the market the next step is to determine the relative demand for each of the three product models. The number of model *j* units sold (D_j) is affected by the mean price of each model averaged over all companies,

$$\bar{p}_j = \frac{\sum_{i=1}^{n} p_{ij}}{nC}$$

and the total advertising for each model,

$$\bar{a}_j = \frac{\sum_{i=1}^{n} \bar{a}_{ij}}{C}$$

Using similar factors to those which determine the total market demand, the final fraction of the market for each model is then determined as a proportion of the total demand,

$$f_j = \frac{\pi_{.j}\alpha_{.j}}{\sum_{j=1}^{m} \pi_{.j}\alpha_{.j}}$$

where the factors are defined thus.

Model Mix Price Factor

$$\pi_j = 6e^{-\kappa_j \bar{p}_{.j}}$$

Model Mix Advertising Factor

$$\alpha_j = \lambda_j \left[\tanh\left(\frac{a_{.j}}{5.0\times10^5} \right) - 0.1\left(1 + \tanh\left(\frac{a_{.j}}{5.0\times10^5} - 2.0 \right) \right) \right]$$

The different demands for each of the products is controlled with the vectors $\hat{\kappa}$ and $\hat{\lambda}$. The values finally chosen are shown below but could be modified to reflect variations in differential price and advertising sensitivities.

$$\hat{\kappa} = \{0.15, 0.11, 0.08\}$$

and

$$\hat{\lambda} = \{0.8, 1.0, 1.2\}$$

Thus total demand for model *j* is given by,

$$D_j = f_j D$$

Step 3: Company Shares of Total Market.

The distribution of each model's demand between companies is controlled by the following six factors, where the subscript *i* specifies the company and *j* the model.

Company Advertising Factor

$$\Psi_{aij} = \tanh\left(\frac{\bar{a}_{ij}}{6.0\times10^4}\right) - 0.1\left[1.0 + \tanh\left(\frac{\bar{a}_{ij}}{6.0\times10^4} - 2.0 \right) \right]$$

Company Salary Factor

$$\Psi_{sij} = 10.57\left[1 - \tanh\left(2.0\times10^{-5} s_i - 1.0 \right) \right]$$

Company Price Factor

$$\Psi_{pij} = 6e^{-0.11 p_{ij}}$$

Company Sales Representatives Factor

$$\Psi_{rij} = 1 - e^{-0.0033 r_i}$$

Company Commission Factor

$$\Psi_{cij} = (1 - e^{-0.23 c_{ij}})$$

where commission (c_{ij}) is expressed as a percentage.

Company Research and Development Factor

$$\Psi_{dij} = \rho\Psi'_{dij} + 2.0\left(1 - e^{-1.3\times10^{-5}\Re}\right)(1-\rho)(0.7 + 0.6U_{0-1})$$

where Ψ'_{dij} is the value of the research and development factor for the previous quarter, ρ is a constant between zero and one representing the rate at which R&D effectiveness decays and U_{0-1} is a uniformly distributed

random number also in the range zero to one. This randomisation models the essentially hit and miss nature of research outcomes. $\Re$ is the dollar amount spent on research and development for the quarter.

The demand to company *i* for product *j* is,

$$d_{ij} = \frac{D_{.j}\Phi_{ij}}{\sum_{i=1}^{n}\Phi_{ij}}$$

given,

$$\Phi_{ij} = \left(1+\frac{\Psi_{aij}}{\sum_{i=1}^{n}\Psi_{aij}}\right)^{w_{aj}} \times \left(1+\frac{\Psi_{pij}}{\sum_{i=1}^{n}\Psi_{pij}}\right)^{w_{pj}} \times \left(1+\frac{\Psi_{rij}}{\sum_{i=1}^{n}\Psi_{rij}}\right)^{w_{rj}} \times \left(1+\frac{\Psi_{sij}}{\sum_{i=1}^{n}\Psi_{sij}}\right)^{w_{sj}} \times \left(1+\frac{\Psi_{cij}}{\sum_{i=1}^{n}\Psi_{cij}}\right)^{w_{cj}} \times \left(1+\frac{\Psi_{dij}}{\sum_{i=1}^{n}\Psi_{dij}}\right)^{w_{dj}}$$

The $w_{aj}, w_{pj}, \ldots$ are weighting factors that allow the effect of each of the $\Psi_{aij}, \Psi_{pij}, \ldots$ to have different magnitudes. The values used in DIMSIM are presented below in Table 1.

	Model 1	Model 2	Model 3
Price	10	3	0.5
Advertising	5	3	3
Sales Reps	2	3	5
Base Salary	2	3	-2
Commission	3	3	4
R and D	2	4	6

Table 1 Weighting Exponents Used in DIMSIM for the Final Demand Equation

Step 4: Unsatisfied Demand

If any company receives more orders than available stock there is no back ordering. The unsatisfied demand is distributed to the other companies in proportion to their original demand for that model. Thus if the demand for product *j* from company ε ($d_{\varepsilon j}$) exceeds their inventory ($I_{\varepsilon j}$) then the quantity of excess orders passed on to company *i* is,

$$\delta_{ij} = \frac{(n-1)(d_{\varepsilon j} - I_{\varepsilon j})d_{ij}}{\sum_{\substack{i=1 \\ i\neq\varepsilon}}^{n} d_{ij}}$$

FUTURE IMPROVEMENTS

DIMSIM has been used successfully over several semesters and feedback from participants suggests that it has a positive effect on their learning outcomes. Over that time the market model described here has proven robust and ostensibly realistic to participants. However, there are several areas where it could be improved, some of which are briefly discussed below.

The influence of price on total market size is less than ideal because it uses industry-wide price averages which may be distorted by participants using extreme pricing, perhaps as an endgame strategy. A possible solution would be to use the harmonic rather than arithmetic price mean which reduces the effect of extreme values (Gold and Pray 1990).

$$\overline{P}_k = \frac{nmC_k}{\sum_{i=1,j=2}^{n,m}\frac{1}{p_{ijk}}}$$

However, further investigation is required and other approaches such as probability distributions may be more appropriate.

There is also no interdependence in the model between demand elasticities for the various decision variables, which is not necessarily realistic. The impact on demand of doubling the advertising budget will not necessarily be the same for products priced at the low end of the market as for those with comparatively high prices but the model presented here cannot reproduce that effect.

The model also takes a fairly simplistic approach to the handling of unsatisfied demand. Rather than simply redistributing it amongst those companies able to supply there should be a more sophisticated mechanism that includes the ability to support back ordering and takes into account the impact on future demand of each company's service level performance.

An alternative approach would be to have a market model that is stochastic in nature where decision variables influence the underlying probability distributions which are used to generate demand. Such a system would make it possible to generate data down to the individual transaction level which would make possible the creation of simulations tailored for a variety of Information Systems courses such as Database Management, Decision Support Systems and Business Intelligence, Systems Analysis and Design, Enterprise Resource Planning and Human Computer Interaction.

REFERENCES

Burgess, T. F. 1991. "The Use of Computerised Management and Business Simulation in the United Kingdom." *Simulation and Games*, v22, n2, pp174-95.

Cotter, R. V. and Fritzsche, D. J. 1995. *The Business Policy Game.* 4th Ed., Prentice Hall, Englewood Cliffs, USA.

Chu C.-H., Huttenstein, M.P. and Greenlaw, P.S. 1996. *Prosim III for Windows.* McGraw Hill.

Faria, A. J. 1987. "A Survey of the Use of Business Games in Academia and in Business." *Simulation and Games*, v18, pp207-24.

Faria, A. J. 1989. "Business Gaming: Current Usage Levels." *Journal of Management Development*, v**8**, pp58-65.

Gold, S. C. and Pray, T. F. 1983, "Simulating Market and Firm Level Demand – A Robust Demand System." *Developments in Business Simulation & Experiential Learning*, v10 pp101-6.

Gold, S. C. and Pray, T.F. 1990, "Chapter 8: Modelling Demand in Computerized Business Simulations." *Guide to Business Gaming and Experiential Learning.* ABSEL, pp117-38.

Kamp, P. V. and Kaiser, H. M. 1999. "Optimal Temporal Policies in Fluid Milk Advertising." *NICPRE Quarterly*, v5 n1, pp1-3.

Keys, B. and Bell, R. 1977. "A Comparative Evaluation of the Management of Learning Grid Applied to the Business Policy Learning Environment." *Journal of Management*, v3 pp33-9.

Lawrence, P.J. 1997. "Business Simulations: Dynamic Computer Based Case Studies for Management Development." *The Place of Information Technology in Management Education* (IFIP WG3.4). Ed. Barta, B., Tatnall, A. and Juliff, P., Chapman and Hall, pp128-36.

Ledman, R. E. 2005. "Student Expectations of Simulations." *Developments in Business Simulation & Experiential Learning*, v32 pp183-5.

McKenna, R. J. 1991. "Business Computerised Simulation: the Australian Experience." *Simulation and Games*, v22, n1, pp36-62.

Murff, E. J. T., Teach, R. D. and Schwartz, R. G. 2006. "Three-Attribute Interrelationships for Industry-Level Demand Equations." *Developments in Business Simulation & Experiential Learning*, v33 pp213-8.

Perotti, P. and Pray, T. 2000. "Visual Modeling of Business Simulations." *Developments in Business Simulation & Experiential Learning*, v27 pp34-41.

Wolfe, J. 1975, "A Comparative Evaluation of Experiential Approach as a Business Policy Learning Environment." *Academy of Management Journal*, v18 pp442-52

AUTHOR BIOGRAPHY

PETER LAWRENCE was born and educated in Melbourne, Australia where he completed a bachelor's degree in Material Engineering and a PhD in statistical, solid-state thermodynamics at Monash University. He has worked as an academic at several Australian universities as well as holding various industry positions as an engineer, research scientist, analyst and consultant to organisations including as ABB, Credit Suisse First Boston, BHP, Ford, Telstra, Ford and the CSIRO in Australia and Europe.

After eight years with School of Business Systems at Monash University he has recently taken up the position of Associate Professor and discipline leader for Information Systems in the School of Business and Informatics at the Australian Catholic University.

A SIMULATION FOR EXPERIENTIAL LEARNING IN SUPPLY CHAIN MANAGEMENT

Peter Lawrence
School of Business and Informatics
Australian Catholic University, St. Patrick Campus
Victoria 3065 AUSTRALIA
Email: peter.lawrence@acu.edu.au

Ann Lawrence
Graduate Business School
Deakin University
Victoria 3144 AUSTRALIA
Email: adkl@deakin.edu.au

KEYWORDS

Education, Decision Making, Simulators, Graphics.

ABSTRACT

Research has shown that putting course content into a realistic context enhances learning and improves business students' enjoyment and understanding of course material. In particular, simulation exercises are a way of giving both real-world context and complexity to problem solving techniques. This paper describes the use of simulation to aid student motivation and understanding of quantitative supply chain and inventory management techniques.

Participants were required to develop operating policies to maximise company profit in a simulated, complex international manufacturing and distribution network by specifying decision variables such as production rates, product mix, inventory policies and transportation arrangements. The simulation demonstrated, though animation, production and financial reporting, the consequences of implementing those policies.

Feedback from the pilot implementation of the exercise was positive.

EXPERIENTIAL LEARNING

Employers are now demanding graduates who have the ability to function in complex business environments. This requires the training and development of graduates with an integrated understanding of how business works, who can work effectively in organizations and apply conceptual knowledge to solve problems (Payne and Whittaker, 2005). Experiential learning using simulations offers assistance though active engagement of the learner. Experience is,

> *" ...not just an observation, a passive undergoing of something, but an active engagement with the environment, of which the learner is an important part"* (Boud, Cohen and Walker, 1993 p 6).

Experiential learning theory defines learning as:

> "*…the process whereby knowledge is created through the transformation of experience. Knowledge results from the combination of grasping and transforming experience*" (Kolb 1984, p. 41).

It allows the learner to be involved fully in the new experience, observe and reflect on these experiences, create concepts that integrate their observations into sound theories and use these theories to experiment, make decision and solve problems. Through continuous experience the learner progresses to higher levels in learning. Experiential learning facilitates the ability to use higher order cognitive skills to solve problems and take decisions (Feinstein Mann, & Corsu, 2002).

INVENTORY SIMULATION

Simulation has been widely used as a mechanism for putting business management decisions into a realistic context for teaching both strategic and operations management in Australia (McKenna, 1991), the U.S. (Faria, 1987) and Britain (Burgess, 1991). Simulation exercises have been used in wide variety of industries and functions including manufacturing, marketing, portfolio management, accounting, human resources and risk assessment (Chu, Huttenstein and Greenlaw 1996; Preismayor 1992; Lawrence and McDonald, 1995; Fripp, 1994; Cotter and Fritzsche 1991).

Computer simulations provide an effective way of teaching business through experiential learning. They allow individuals to immerse themselves in the experience and "*learn by doing*" (Senge, 1990); to make decisions in complex, dynamic and uncertain environments and provide opportunities to practice decision making skills (Palia, 1989).

Simulations are useful in the acquisition of higher level skills (Wolfe, 1997). They enable participants to apply

theoretical knowledge and permit reflection consistent with Kolb's theory of learning. Participants can observe the implications of their decisions.
Simulations also empower participants in the learning process, encourage interaction with others and enhance involvement and interest, which leads to increased motivation and meaningful learning (Payne and Whittaker, 2005). They provide an alternative teaching method and adjunct to traditional case studies and lectures.

Making the simulation user friendly and fun is an important consideration in increasing the levels of engagement with the task (Wynder, 2004). When the task is a source of interest, enjoyment and personal challenge this can result in intrinsic motivation (Amabile, 1997). However extrinsic motivators, that is the motivation to engage in an activity to meet some imposed requirement or receive some reward or recognition that is external to the task itself, are also required.

The exercise described here was designed to facilitate learning in the area of international supply chain management. The simulation itself was developed in Arena and was very tightly coded, using low-level SIMAN objects. This low-level coding minimises the source code size and enables the model to be run, despite its relatively high complexity, using the freely available academic version of the Arena software. Thus the model can be installed and run on any university or student owned computer.

The Exercise

The object of the exercise is to present students with a management situation requiring quantitative analysis but with a degree of complexity well above that of traditional textbook problems. In this way the techniques used are placed into a more realistic context which highlights both their utility and limitations. It is based around a hypothetical company manufacturing a luxury item at their factories in China and South America. Each factory can run up to ten production lines each of which can be tooled to construct any one of the three different available models of the product. After production, products are stored in warehouses local to each factory until they are required for distribution.

International distribution is carried out through a series of warehouses in Genoa, Melbourne, Los Angeles, Singapore and Tokyo, which are used to distribute to the local dealer networks. Transport between factory and distribution warehouses is primarily by sea for which the company charters a number of cargo vessels.

Students are required to develop supply chain management policies to maximise company profit over the remaining product life cycle by manipulating various decision variables including:

- The number of production lines producing each model at each of the company's factories,
- The reorder quantity for each model at each distribution warehouse,
- The minimum stock level reorder point for product replenishment of each model at each of the distribution warehouses,
- Which of the two factories to order product from on a case by case basis,
- The time, towards the end of the products' life cycle, to continue production runs before tooling set up for production of the new product,
- The date after which to stop reordering product for the distribution warehouses in anticipation of the new product's release,
- Shipping procedures for product from factory to distribution warehouse.

In making their decisions students need to consider a complex array of factors that affect the profitability of operations. These factors include:

- Production costs and overheads,
- International time zone differences,
- Inventory holding costs which vary between locations,
- Dealer demand for product, which varies for each model around the world,
- Product mix demand, which varies from country to country,
- Lead times resulting from production schedules,
- Lead times required for international transport,
- Variability of transport times due to weather variations and customs delays,
- Shipping costs including crews, fuel charter and port fees,
- Product life cycle and the disposal of dead product stock at the end of its life cycle,
- Lost orders due to excessive back ordering and delivery delays,
- Details of customer demand at the distribution warehouses in terms of the frequency and sizes of orders received,
- Wholesale price of goods sold from each warehouse,
- Estimates of the expected inventory level at each factory and warehouse at the commencement time of the simulation,
- The anticipated value of the dumped stock of old product.

Having made their policy decisions the simulation is run and students are able to see their effects via both animation and a series of company reports. Animation demonstrates the progress of company operations including the movement of product (Figure 1) and the inventory levels and backorder status at each factory (Figure 2) and distribution warehouse (Figure 3).

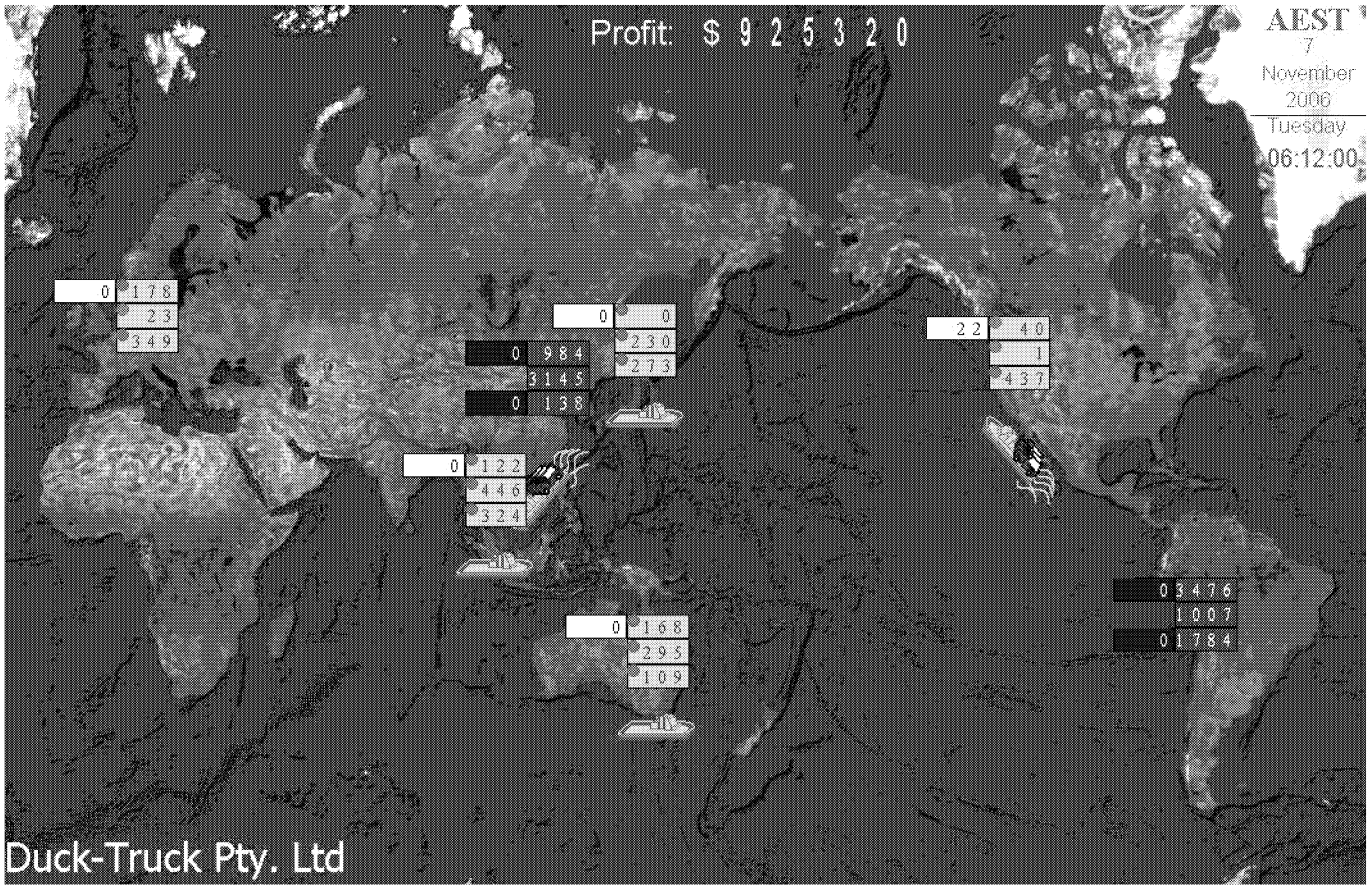


Figure 1 The Main Animation Screen for the Simulation.

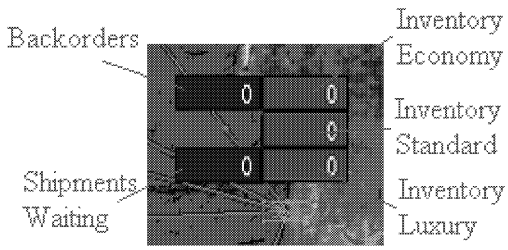


Figure 2 Data Displays at Factory Sites.

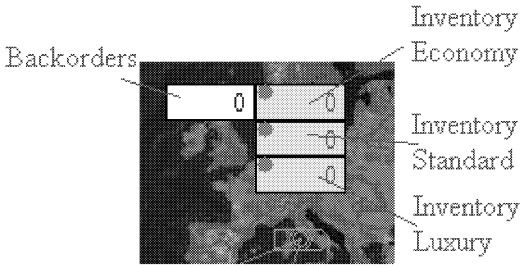


Figure 3 Data Displays at Warehouse Sites.

More detailed information on company operations is made available through various operational reports in the form of plain text and formatted Excel files. Figure 4 and Figure 5 show some sample extracts from these reports. Details reported include:

- The total number of customer orders received at each distribution warehouse,
- The total number of units of each model sold by each distribution warehouse,
- Statistics relating to the time customers had to wait for their orders to be filled and the number of backorders created,
- The estimated number of orders lost due to excessive build up of backorders,
- The number of orders received and shipped by each factory from each distribution warehouse,
- Statistics relating to the time taken to fill orders and the number of backorders required,
- Statistics relating to the time shipments spent waiting on the docks and the number of shipments waiting,
- Factory production volumes,
- Statistics relating to inventory levels at both the factories and the distribution warehouses around the world,
- Details of shipping activity including trips made and associated costs,
- Reorder lead times experienced by each distribution warehouse from each of the factories,
- A breakdown of revenue and costs including the final profit for the duration of the simulated management period.

```
**********************************

*****    Customer Orders     *****

**********************************

           Genoa     Los Angeles   Melbourne   Singapore    Tokyo
            799          920         1655        1433        648


**********************************

***** Warehouse Sales Report *****

**********************************

           Genoa     Los Angeles   Melbourne   Singapore    Tokyo
 Economy   1151         2565         1431        3218          0
 Standard  2773         5185         1176        1009        390
 Luxury     876         2497          714         215       2151


**********************************

***** Customer Waiting Time  *****

**********************************

Identifier                                Average    Half Width  Minimum    Maximum    Observations
-------------------------------------------------------------------------------------------------

Customer Wait Genoa                       3.7799     5.1046      .00139     103.07         799
Customer Wait Los Angeles                 4.2064     1.7995      .00139     30.027         920
Customer Wait Melbourne                   .00291     1.1896E-04  .00139     .01042        1655
Customer Wait Singapore                   .56225     .47192      .00139     17.018        1433
Customer Wait Tokyo                       .42517     (Corr)      .00139     13.013         648


**********************************

*****       Backorders       *****

**********************************

Identifier                                Average    Half Width  Minimum    Maximum    Final Value
-------------------------------------------------------------------------------------------------

Waiting Customers Genoa                   5.5005     (Corr)      .00000     21.000      1.0000
Waiting Customers Los Angeles             8.0187     2.2940      .00000     22.000      20.000
Waiting Customers Melbourne               .00669     7.7871E-04  .00000     14.000      .00000
Waiting Customers Singapore               1.4657     1.1091      .00000     22.000      .00000
Waiting Customers Tokyo                   .50102     .55086      .00000     19.000      .00000
```

Figure 4 Extract from Aggregate Output Report

Replication Number	Sim. Time (Days)	Shipped From	Shipped To	Units			Run Time
				Economy	Standard	Luxury	
1	34.17	Shanghai	Los Angeles	400	0	400	Tue Aug 14 13:59:01 2007
1	37.24	Valparaiso	Melbourne	0	400	0	Tue Aug 14 13:59:01 2007
1	46.60	Valparaiso	Genoa	0	0	400	Tue Aug 14 13:59:01 2007
1	51.16	Shanghai	Los Angeles	0	400	0	Tue Aug 14 13:59:01 2007
1	70.17	Shanghai	Singapore	400	0	0	Tue Aug 14 13:59:01 2007
1	72.49	Valparaiso	Genoa	0	400	0	Tue Aug 14 13:59:01 2007
1	80.14	Shanghai	Los Angeles	0	400	0	Tue Aug 14 13:59:01 2007
1	105.27	Shanghai	Tokyo	0	0	400	Tue Aug 14 13:59:01 2007
1	106.74	Valparaiso	Los Angeles	0	0	400	Tue Aug 14 13:59:01 2007
1	106.90	Shanghai	Singapore	0	400	0	Tue Aug 14 13:59:01 2007
1	123.35	Valparaiso	Los Angeles	400	400	0	Tue Aug 14 13:59:01 2007
1	128.01	Shanghai	Genoa	0	400	0	Tue Aug 14 13:59:01 2007
1	146.32	Valparaiso	Melbourne	0	0	400	Tue Aug 14 13:59:01 2007

Figure 5 Extract from Detailed Shipping Report

Optimisation

In addition to using the simulation to test the effectiveness of their chosen supply chain management policies, students were then able to use the simulation to search for a better solution using optimisation software. The complexity of the search space meant that convergence could be slow and that any identified profit maxima was likely to be local rather than global. The simulation thus allowed students to investigate issues relating to search space boundary and initial condition selection.

FEEDBACK

The exercise was used in a pilot study conducted using students studying the application of Operations Research techniques in logistics. After the exercise was completed, qualitative feedback was sought from students to assess their perceptions of the learning experience as compared to alternative modes of instruction delivery. Positive student feedback gave strong indications of the usefulness of experiential exercise in achieving the initial aims of the course. In particular students identified various perceived benefits as compared to purely theoretical studies of operations management techniques. These included:

- An appreciation of the complexity of real operational decision making,
- An appreciation of the complexity of the relationships in the policy decisions in the simulation,
- An appreciation of the simplifications and limitations inherent in traditional inventory management techniques,
- The importance of efficient optimisation algorithms in complex, multi-dimensional decision spaces,
- The role simulation can play in dealing with system complexity beyond the scope of most analytical Operations Research techniques,
- Increased enjoyment of the simulation method over traditional teaching methods leading to greater intrinsic motivation,
- The requirement of the task to maximise profit led to increased extrinsic motivation.

CONCLUSIONS

The interactive computer inventory simulation was rated as effective as a learning experience in assisting students to apply the principles and concepts of operations research. It was also deemed effective and enjoyable as an alternative to traditional methods of instruction in this discipline. However, as this was an informal pilot study, further extensive research, employing statistical analysis techniques, is required to more rigorously evaluate the learning experience, alternative methods of instruction and levels of overall satisfaction with the simulation.

REFERENCES

Amabile, T. M. 1983. "The Social Psychology of Creativity: Componential Conceptualizations", *Journal of Personality and Social Psychology,* v45, n2, pp357-76.

Boud, D., Cohen, R. and Walker, D. 1993. "Introduction: Understanding Learning from Experience". In D. Boud, R. Cohen and D. Walker (eds). *Using Experience for Learning*, pp1-17. New York: Nichols Publishing Company.

Burgess, T. F. 1991. "The Use of Computerised Management and Business Simulation in the United Kingdom", *Simulation and Games*, v22, n2, pp174-95.

Cotter, R. V. and D. J. Fritzsche. 1991. *The Business Policy Game*, 3rd Ed., Prentice Hall, Englewood Cliffs, USA.

Chu C-H., Huttenstein, M.P. and Greenlaw P.S., *Prosim III for Windows*, McGraw Hill (1996)

Faria, A. J. 1987. "A Survey of the Use of Business Games in Academia and in Business." *Simulation and Games*, v18, pp207-24.

Feinstein, A. H., Mann, S., and Corsun, D. L. 2002. "Charting the Experiential Territory". *Journal of Management and Development,* v21, n10, pp732-744

Fripp, J. 1994. "Why Use Business Simulations", *Executive Development*, v7, pp29-32.

Kolb, D. A. 1984. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall. New Jersey, USA.

Lawrence, P.J. and McDonald, T. 1995. "The General Insurance Management Instructional Computer Simulation", *Proceedings of ASCILITE 1995*, Dec. 1995, Melbourne, Australia, pp 319-25.

McKenna, R. J. 1991. "Business Computerised Simulation: the Australian Experience", *Simulation and Games*, v22, n1, pp36-62.

Palia, 1989. "Sensitivity Analysis with the COMPETE IFPS/Personal Student Analysis Package: A Marketing Decision Support System", *Developments in Business Simulation and Experiential Learning*, v16, pp141-144.

Payne, E. and Whittaker, L. 2005. "Using Experiential Learning to Integrate the Business Curriculum", *Developments in Business Simulations and Experiential Learning*, v32, pp245-254.

Priesmeyer, H. R. 1992. *Strategy! A Business Unit Simulation*, 2nd Ed., South-Western Publishing Co., Cincinnati, USA.

Senge, P.M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Sydney: Random House.

Wolfe, J. 1997. "The Effectiveness of Business Games in Strategic Management Course Work", *Simulation and Gaming*, v28, n4, pp360-76.

Wynder, M. 2004. "Facilitating Creativity in Management Accounting: a Computerized Business Simulation", *Accounting Education*, v13, n2, pp231-250.

AUTHOR BIOGRAPHIES

PETER LAWRENCE was born and educated in Melbourne, Australia where he completed a bachelor's degree in Material Engineering and a PhD in statistical, solid-state thermodynamics at Monash University. He has worked as an academic at several Australian universities as well as holding various industry positions as an engineer, research scientist, analyst and consultant to organisations including as ABB, Credit Suisse First Boston, BHP, Ford, Telstra, Ford and the CSIRO in Australia and Europe.

After eight years with School of Business Systems at Monash University he has recently taken up the position of

Associate Professor and discipline leader for Information Systems in the School of Business and Informatics at the Australian Catholic University.

His research interests include simulation and modelling in manufacturing, business process design and financial analysis.

ANN LAWRENCE is a Senior Lecturer and subject leader of Human Resource Management in the Graduate Business School and co-director and facilitator of the China International Study Program at Deakin University in Melbourne, Australia.

Ann's research interests include value congruence as predictors of commitment; strategic human resource development and experiential learning; corporate - university alliances; and repatriation support practices.

She has taught a variety of HR subjects in both undergraduate and postgraduate programs in Australia and overseas including China, Hong Kong, Malaysia, and Singapore. She has consulted widely to private industry in the field of HRM organisational culture and values, leadership and change and has worked in human resource development in Australia and in Switzerland.

Ann is a Certified Member status of the Australian Human Resources Institute, and a Member of the International Society for the Study of Work and Organizational Values. She has extensive experience in curriculum development, including executive leadership and development, HRM, Managing Change, Organisational Behaviour, and Management Communications.

BIOLOGICAL SIMULATION

LIFE CYCLES AND COMPETITION IN MODELLING OF ARTIFICIAL AND BIOLOGICAL CONTROL SYSTEM

Ivars Mozga
Uldis Grunde-Zeiferts
Sandis Sudars
Egils Stalidzans
Biosystems Group, Faculty of Information Technology,
Latvia University of Agriculture,
Liela iela 2, LV 3001, Jelgava
Latvia
ivars.m@gmail.com

KEYWORDS
Biological control system, life cycle, competition.

ABSTRACT

The modelling of control loops within functioning biological organisms is a prerequisite for a successful treatment or therapy development and implementation in biological systems. Similarities in emergence and sequential lifecycle steps of human made artificial control systems (ACS) and natural biological control systems (BCS) in multigeneration scale are analysed and compared to extract additional knowledge for modelling procedures.
The development of ACS includes: 1) definition of targets, 2) design of control system, 3) execution of control system, 4) behaviour of technical object as observable result, 5) feedback to the design.
The development of BCS has appropriate steps in different execution and includes: 1) predefined targets (survival and reproduction), 2) genome (as design of control system), 3) cells and organism (as execution of biological control system), 4) behaviour of biological object as observable result, 5) feedback to genome (as design of control system).
The differences and common features in execution principles of each step in both lifecycles are discussed. Consequences of ACS and BCS lifecycle joining in case of artificial control of biological system are discussed.
Perspectives of Systems Biology as emerging science about modelling of biological processes regarding modelling of BCS are mentioned.

INTRODUCTION

Historically the control theory first arose out of a need to understand the behavior of systems. This theory was then used to design and engineer better systems. It is not difficult to foresee that, in biology, control theory may follow the same path (Ingalls et al. 2006). The effect prediction and the analysis of control system implementation becomes complicated in case of biological object of interest: it has own control loops and competition of several loops of object and designed control system can occur. An understanding of dynamic behaviour of biological object is important forecasting consequences of any interaction with it. In the new science Systems Biology (SB) behaviour of biological system can be analysed as expression of genetically coded control system of an organism (Alberghina and Westerhoff 2005, Ingalls et al. 2006, Klipp et al. 2006).
Holistic approach based quantitative model of a biological object or process becomes a tool that can either confirm or deny new hypotheses about elements and interactions of a functioning biological system as model and its original (biological object) can be compared in dynamic behaviour. Validated model can become a tool for intensive experiments before implementation of a control or therapy under most critical circumstances taking into account transition processes.
In multigeneration evolutionary scale the development of technical and biological systems has similarities and differences in strategies and means of execution.
The goal of this paper is to analyse differences and similarities in evolutionary lifecycles in scale of many generations of biological objects as natural biological control systems and human made artificial control systems. Joint life cycles in case of controlled biological object have to be analysed. Control analysis related possibilities of Systems Biology as a new biosystem modelling approach have to be analysed.

DEFINITIONS

An artificial control system (ACS) is a human designed control system. It can be executed by technical, chemical, biological or other means. By ACS in this paper is meant very wide range of control systems for example simple technical control system (climate control system in a building), complex technical control system (control system of an aircraft), human made control system of biological objects (fermentation process control), human designed control of biological object by another biological object (pest control by purposeful introduction of their biological enemies).
A biological control system (BCS) is in biological reproduction process developed control system that ensures internal processes within biological object and interaction processes with environment. Features of biological objects are metabolism and reproduction. BCS controlled biological objects are for example all living organisms (plants, animals, humans) as well as their subsystems (body temperature control, metabolism, processes within a cell).
There are different definitions of Systems Biology (SB) available indicating different accents of SB mission:

- SB research behaviour and interactions of elements in functioning biological system (Palsson 2000, Ideker et al. 2001).
- SB can be defined as understanding of complex biological systems integrating experimental and computational research (Kitano, 2002).
- SB studies how properties of live forms arise from interaction of their components (Reiss 2005).

Systems biology approaches comprise (Reiss 2005):

- The enumeration of network and pathway components in complex biological contexts,
- The reconstruction and mathematical modelling of networks, pathways or living systems,
- The mathematical representation of networks based on quantitative biological datasets,

The mathematical analysis and simulation of networks to assess their properties and biological experiments to verify or falsify mathematical models of biological systems.

ACS AND BCS LIFECYCLES

Successful ACS and BCS have to fulfil the goal of control. ACS is made by humans and therefore the lifecycle of successful control system is well defined. Experience and common sense set rational sequence of operations.
The lifecycle of ACS and BCS consist of operations with similar meaning executed by completely different methods and means (Figure 1).

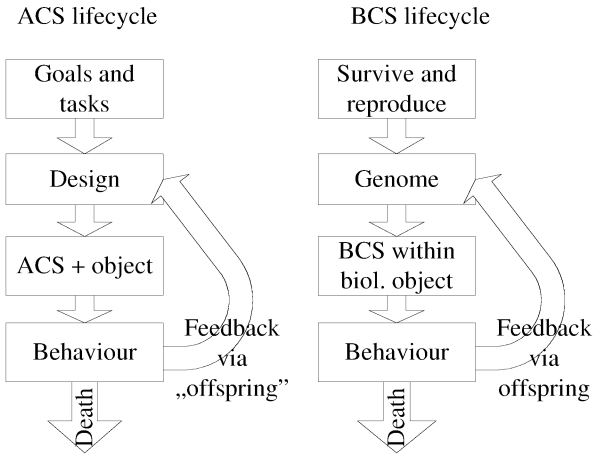


Figure 1. Life cycle of ACS and BCS.

Goal setting

The first stage of ACS lifecycle is the setting of control goal by developers of system - humans. Different tasks can be set to reach goals: stability, safety, efficiency and others.
The first stage of BCS lifecycle – setting of goals and targets - takes place without humans. Still there are two targets that are common for all biological objects: reproduction and survival as there is no other way to continue evolution and other criteria's become less important in multigeneration scale.

Design

The second stage of ACS lifecycle is the development of a design to reach the tasks. The design usually is based on some of the best previous generation designs that have to be more or less improved accordingly to collected experience.
The second stage of BCS lifecycle ends by next generation organism design: genome (Klipp et al. 2006) represented as a set of DNA double helix. That is a form of design record that contains all the necessary information to build a biological system. A incremental development of genome takes place in iterations with one generation step. The base of the new genome is the combination of parent genomes that have proven ability to survive and reproduce. The next iteration of a genome is modified by crossover and mutations thus providing flexibility of next generation. A set of the BCS (genome) of every specie of biological organism is recorded by the same means – a DNA molecule thus indicating enormous potential of this form of record.

Implementation

ACS has to be built accordingly its design. Often the object to be controlled is an autonomous system and ACS is a separate unit. Build-up of ACS usually is modular. Ready modules are interconnected until stage when system can function.
The implementation of BCS design is automatic reproduction of cells the way as it is set in the genome. BCS develop simultaneously with the biological object (from system of several cells to complete organism) to be controlled and has to control the system during different development stages where different modules develop from stem cells. In case of BCS it is usually impossible to split the object and BCS.

Behaviour

The behaviour of the ACS in a technical system should correspond to the planned dynamic behaviour. ACS has high repeatability: Properly functioning ACS behaves the same way if circumstances stay the same. Complicated control system often needs fine-tuning or some not automated adjustments.
The genome defines BCS behaviour in all its complexity. Even flexibility is encoded in the genome as it does not change during a lifetime of a single organism. Some control loops of BCS that are responsible for internal processes in biological object act since birth through different growth stages of biological object. Parents and specific circumstances of the environment tune later some other loops (conditional reflexes) of BCS that are responsible for interactions with environment where the biological object has to fulfil its targets – survive and reproduce. This kind of flexibility enables adaptation of biological object where it is necessary. BCS has low repeatability: BCS may behave different even if circumstances stay the same.

Feedback

The fine-tuning and some adjustments as described earlier in the item "Behaviour" may be part of multigeneration level behaviour of ACS and BCS.
ACS is successful if the goal of implementation is reached. Then ACS should be copied (reproduced) accordingly previous successful design creating a batch of ACS as a "offspring". The design might be slightly corrected for specific circumstances if they are known in advance. In the case of un-

successful ACS design can be radically changed in one generation step using other means or (and) structure of ACS.
BCS is successful if readiness for reproduction is reached and an offspring is produced. Otherwise the genome (design) of BCS will not be repeated in the next generation. The research in genetics gives overview of genome repeatability in the next generation (Klipp et al. 2006). Actually it is not repeating. The crossover and mutations create a variety of BCS offspring preparing designs of BCS for possibly changing circumstances. Still the main part of offspring is almost a copy of successful previous generation. Mutations take care of extremes in the new generation. That is a strategy that allows being ready for unpredictable environment changes, which are even not experienced by previous generation. Still the feedback of successful parents is a strong guideline.

ARTIFICIAL CONTROL OF BIOLOGICAL OBJECTS

The interest about biological objects and processes if often caused by a wish to control them (change their behaviour). Control of biological system without artificial changing BCS (genome) can be executed by a modification of environmental parameters of the biological object. BCS as a set of control loops will react to changes in the environment. The knowledge about reaction rules of BCS (including dynamics) to environmental changes is the key to ACS design. If the goal of ACS implementation is a specific behaviour of biological object it is necessary to know which environmental parameters can cause it. ACS has to be designed to ensure the necessary environment.
ACS can be optimised accordingly efficiency criteria (costs, safety, environmental friendliness) if there is more than one set of environmental parameters that cause needed behaviour of BCS.
Thus the development of efficient ACS of a biological object is closely related to the BCS. Lifecycles of ACS and BCS (Figure 1) have to come together into a unit "artificially controlled biological object" (Figure 2). Assuming that we can not change BCS lifecycle, the one of ACS has to be adapted.
Several significant changes take place (Figure 2):

1. Two control systems: BCS and ACS are working in parallel. They interact and compete via one or several environmental parameters. Transition processes (Weyrick 1975, Stalidzans 2005; Stalidzans and Markovitch 2005a,b; Dorf and Bishop 2005) become critical. Only ACS can be modified to ensure safe control process. This kind of task can be solved by the methods of control theory if both control systems are described mathematically by differential equations. That can be done for ACS as it is human made system. Quantitative mathematical description of BCS is problematic as its design (genome) is not human made and the list of system components is not sufficient without description of their interactions. The quantitative description of BCS dynamics is a huge reverse engineering problem where SB offer new approaches and methods trying to combine dynamic experiments of behaviour with DNA-encoded design elements of BCS. The current stage of SB does not offer complete BCS model of even smallest and simplest cell. Still often model of a part of the process of interest is highly valuable.

2. Feedback loop of BCS is changed. Within the feedback loop "Genome", "BCS within biological object" and "Behaviour" the node "BCS within biological object" is changed by interaction with ACS forming new node "ACS+BCS within biological object". As a result in next generation's better reproduction rate will get the biological objects, which are the best for environment modified by ACS. Thus genome (design of BCS) via feedback will shift from optimal behaviour in the environment towards optimal behaviour in the environment changed by ACS. This fact brings potential danger: will biological object survive if ACS is removed and the changed genome (ACS impact) will have to fulfil survival and reproduction target in historical (without ACS) environment?

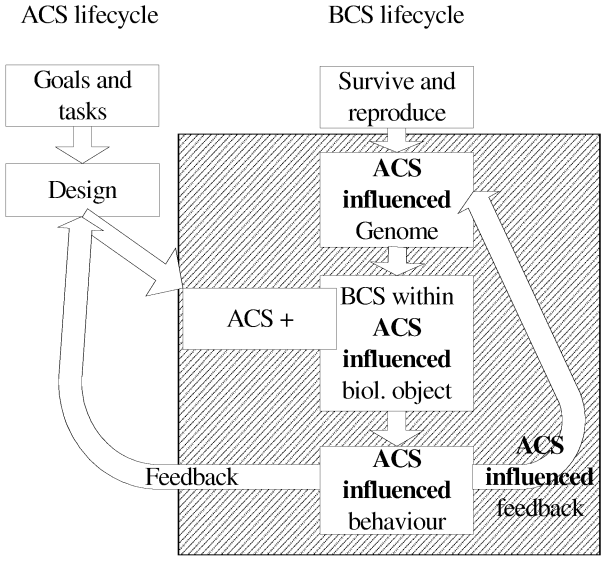


Figure 2. Lifecycle of meld ACS and BCS during control of biological objects. Colored area is impacted process of BCS cycle.

3. A new feedback loop "Design", "ACS+BCS within biological object" and "Behaviour" has developed. Thus design of ACS can be improved taking into account human made and predictable ACS and hardly predictable nature product BCS. The hardly predictable BCS cause hardly predictable behaviour as well as its feedback to the design. Thus ACS lifecycle has to be more adaptive to stay effective in next "generation" of ACS.

4. Overlapping of feedback loops mentioned in items 2 and 3. Both loops have two common nodes: "ACS+BCS within biological object" and "Behaviour". Two feedback loops improve their performance at the same time accordingly the node "behaviour". This is another competition of ACS and BCS taking place in different time scale compared to the competition in dynamics described in item 1). Control system design competition (design in case of ACS and genome in case of BCS) takes place in long term – many generations of ACS and BCS development. Also there transition processes are possible because of competition but it takes place in respectively slow speed. Accordingly to control theory adaptivity of ACS design has to be quicker than the one of genome to take lead in this competition.

PERSPECTIVES OF SYSTEMS BIOLOGY (SB)

The representation of biological models in form of differential equations used in SB is the necessary precondition in use of control theory to describe BCS (regulatory networks in SB) the same way as ACS. Thus SB can contribute in control development of biological objects (Ingalls et al. 2006). Some directions of SB and other related sciences like bioinformatics aim to describe evolutionary process allowing analysis of biological objects in multigeneration scale as well measuring intensity of mutations and crossover and their effects on evolution.

Currently SB is mainly focussing on simulation of biomolecular processes within and between cells that are building blocks of any living organism. This knowledge has to be used in quantitative modelling of genetic, subcellular, cellular, tissue, organ, and system structures (Noble, 2006) that very often are targets of biological object control.

CONCLUSIONS

The differences in evolutionary lifecycles in scale of many generations of biological objects as natural biological control systems and human made artificial control systems are found mainly in means of their execution while the function of feedback loop has many similarities.

The development of ACS includes 1) definition of targets, 2) design of control system, 3) execution of control system, 4) behaviour of technical object as observable result, 5) feedback to the design. The development of BCS has appropriate steps in different execution and include 1) predefined targets (survival and reproduction), 2) genome (as design of control system), 3) cells and organism (as execution of biological control system), 4) behaviour of biological object as observable result, 5) feedback to genome (as design of control system).

Lifecycles of ACS and BCS partially meld in case of artificial control of biological object. Following new effects take place: 1) two control systems (ACS and BCS) act in parallel as some of their loops cross and transition process has to be taken into account, 2) feedback loop of BCS is changed, 3) feedback loop of ACS is changed, 4) feedback loops in ACS and BCS lifecycles overlap. Melding of ACS and BCS development lifecycles cause evolutionary shift in genome that has to be compensated changing ACS.

The new science Systems Biology aims to describe behaviour of BCS controlled biological systems in form of quantitative dynamic models. This approach allows application of automatic control theory for ACS development and optimisation of ACS and BCS collaboration leading to improvement of control development for biological processes.

REFERENCES

Alberghina L., Westerhoff H. (Ed.) (2005) Systems Biology: Definitions and Perspectives, Springer, 408p.

Dorf R.C., Bishop R.H. (2005). Modern Control Systems, Tenth edition. Pearson Education , Inc. Pearson Prentice Hall. 634 p.

Ideker T., Thorsson V., Ranish J.A. et al. (2001) Integrated genomics and proteomic analyses of a systematically perturbed metabolic network. In: Science 292, pp.929-934.

Ingalls B.P., Yi T., Iglesias P.A. (2006) Using Control Theory to Study Biology. In book System Modeling in Cellular Biology, Ed. By Zoltan Szallasi, Jorg Stelling and Vippul Periwal, Massachusetts Institute of Technology. 243-267.

Kitano H. (2002) Computational systems biology, Nature 420, pp.206-210.

Klipp E., Herwig R., Kowald A., Wierling C., Lehrach H. (2006). Systems Biology in Practice. Concepts, Inplementation and Application. WILEY-VCH Verlag GmbH&Co KgaA. 465 p.

Noble D. (2006) Multilevel Modeling in Systems Biology: From Cells to Whole Organs. In book System Modeling in Cellular Biology, Ed. By Zoltan Szallasi, Jorg Stelling and Vippul Periwal, Massachusetts Institute of Technology. 297-312.

Palsson B. (2000) The challenges of in silico biology. In: Nature Biotechnology 18, pp.1147-1150.

Reiss T. (2005) Workpackage 1: International Benchmarking and foresight of systems Biology. The Take-off of European Systems Biology (EUSYSBIO). Karlsruhe, Germany, 44 p.

Stalidzans E. (2005). "Algorithms of computer control of multiobject biological systems." PhD thesis. Riga Technical University, Riga, p. 156.

Stalidzans E., Z. Markovitch (2005 a) "Methodology of control system development for biological systems under information insufficiency". In European Modelling Simulation Symposium EMSS 2005, Marseille, France, 20-22.October 2005. pp.169-175.

Stalidzans E., Z. Markovitch (2005 b). "Development of dynamic model for a biological system under conditions of insufficient information". In Proceedings of ITAFE'05 International Congress on Information Technology in Agriculture, Food and Environment, Adana, Turkey October 12-14, 2005, pp. 337-344.

Weyrick, Robert C. (1975) Fundamentals of automatic control. New York: McGraw-Hill, 397 p.

AUTHOR BIOGRAPHY

IVARS MOZGA studied IT in Latvia University of Agriculture. Since 2006 he is a PhD student in Latvia University of Agriculture specializing in computer control of biological systems. Doctoral studies are connected to modeling of biomolecular processes within cellular structures.

ULDIS GRUNDE-ZEIFERTS studied Information Technologies in Latvia University of Agriculture. Since 2006 he is a PhD student and lecturer in Latvia University of Agriculture specializing in computer control of biological systems. He is mainly interested in recognition of biological control loops and their parameters.

SANDIS SUDARS studied IT in Latvia University of Agriculture defending masters thesis in 2007. Mainly interested in modeling technologies of technical and biological systems and software development.

EGILS STALIDZANS studied in Riga Technical University (Latvia), Faculty of Computer Science and Information Technology where defended his PhD in 2005. Research is mainly related to development of mathematical models under information insufficiency. In 2007 he became assistant professor and senior researcher at Latvia University of Agriculture, Faculty of Information technology. Since 2006 he leads Biosystems Group. He is interested in application of Systems Biology approach in control of industrial bioprocesses.

Mitochondrial OxydoReduction Simulation using Multi-Agent System

N. Parisey
J.P. Mazat
Physiopathologie Mitochondriale,
INSERM - U688 - Univ. Bordeaux ,
parisey@labri.fr

M. Beurton-Aimar

LaBRI
UMR5800

September 28, 2007

ABSTRACT

Since twenty years, more and more crystallographic structures of enzymatic macromolecular complexes are available in international data banks. Movements of large protein structures are key components in ligand docking and enzymatic catalysis. Simulations of enzymatic reactions must take into account such structural movements. Our aim is to combine modeling of the redox reactions and modeling of the conformational changes of enzymes structures in order to describe the dynamical functioning of redox enzymes. An agent based system has been developed to simulate movements and redox reactions. We have applied our approach to the complex III of the mitochondrial respiratory chain. Using this model, we were able to assess quantitative and qualitative enzymatic kinetic behaviours like conversion rate of the overall reaction, individual electrons path within the complex and short-circuits

Keywords : redox reaction, agent based simulation.

Introduction

Multi-Agents Systems (MAS) are currently used to investigate structures, behaviours and functions of metabolic processes [PB04] [KMM+03] [WW06]. Metabolic processes involve interactions of different entities, molecules and enzymes, within a situated physical environment. A metabolic process is a set of one to many biochemical reactions, for example the redox reactions. These biochemical reactions are discrete and local events that will create global behaviour and organisation within the metabolic system. In silico, simulations of such reaction chains can help understanding normal or pathological cell behaviours.

Mitochondria are intra-cellular organelle in eukaryotes. Each cell is filled with a set of hundreds of mitochondria. One of the main role of mitochondria is energy generation for the cell. The energy delivered by mitochondria is the result of a complex mechanism, involving a series of redox reactions called *"Respiratory Chain"* which uses the oxygen we breathe in and the nutriments we ingest.

The main goal of this work is to simulate the internal reactions of redox enzymes using an agent based system in order to uncover the possible short-circuits and deleterious mechanisms that could lead to pathological situations. The following sections present, firstly, the modeling of internal macromolecular movements and redox reaction rates for redox enzymes, then the agent based model we have designed, followed by an application to a particular redox enzyme. Finally, we discuss future works.

Assessing macromolecular internal movements and reaction rates

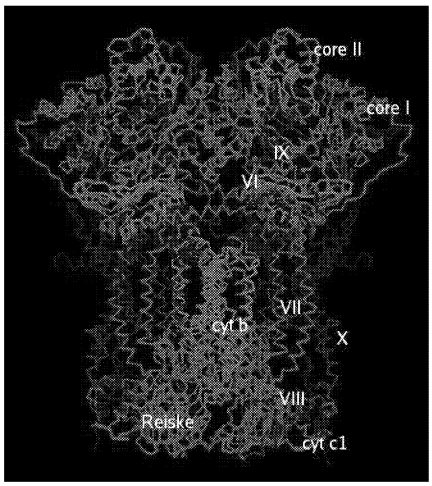


Figure 1: A typical redox enzyme: the complex III

Figure 1, generated with [SMW95], represents a typical redox enzyme with its different biological subunits and its thousands of atoms. The corresponding crystallographic structure was taken from the `Protein Data Bank` [BKW+77]. It exists many techniques in molecular modeling to uncover the possible internal movements

of such macromolecule. We have chosen to rely on geometric techniques in order to partition and to define internal movements of redox enzymes. First, we have split the enzyme into rigid bodies which can be link with hinges. Then, within each part of the rigid bodies, we have defined small regions of interest and we have used two other algorithms to find their movements.

Movements of large enzyme domains

In order to partition the redox enzyme into an articulated rigid body, the `Hingefind` algorithm [WS97] uses two crystallographic structures of the same molecule. The algorithm is a two pass algorithm:

- It detects the rigid domains within the molecule by finding for each domain a translation vector and a rotation matrix. The translation vector and rotation matrix enable a set of atoms from crystallographic structure A to superimpose with the same set on crystallographic structure B within a certain margin error based on crystals resolutions.
- Then, it uses the vectors and matrices to define hinges (that is axis and angles) between previously defined domains.

Internal movements within large enzyme domains

To access movements of small subset of atoms within each part of the previously generated rigid body, we have used a specialized 'Pebble Game' [DM95] algorithm called `FIRST` [JRKT01] that takes into account physical and chemical constraints like covalent and hydrogen bonds between atoms. In this algorithm, the macromolecule is seen as a graph. Vertexes are atoms linked together by edges representing physical bonds. The edges are tagged and thus ranked by how strong the physical bonds are between atoms. Pebbles are assigned to vertexes. Pebbles represent degree of freedom in a 3D space. They will be exchanged between vertex during the algorithm run. This will partition the macromolecule, or part of it in our case, into rigid and flexible subsets.

A second algorithm, called `FRODA` [SWT05], is then used to access movements possibilities for the newly partitioned structure. This algorithm will randomly move subset of atoms (the subsets being based on the `FIRST` partitioning) and see at each step that physical constraints (non overlapping atoms, minimum energy conformation for the whole structure) are respected. Thus, it gives us possible motions for the small subsets of atoms we seek. In a redox enzymes, those subsets of atoms will be the one linked with redox reactants, mainly iron atoms clustered into the macromolecule. Having the movement of these iron atoms will enable us to find the variation in electron transfer rate between them during the redox enzyme work.

Redox reaction rates

Dutton and Moser have predicted the electron transfer rate between two redox molecules or two redox centers [DM94] (that is molecules clustered within a macromolecule). They use a phenomenological equation that exponentially link the transfer rate to the distance (in Angstrom) between redox centers.

$$log_{10}(ke) = 15 - 0.6 * R - 3.1 * \frac{(\Delta G^0 + \lambda)^2}{\lambda}$$

where:

- ke is the transfer rate in electron(s) per second,
- R is the distance between two redox centers,
- ΔG^0 is the free energy of the redox system,
- and λ is the reorganisation energy of the system.

We used this equation to get probabilities of electron transfer between two redox centers during a time step of our simulations.

Stochastic simulation using an agent based model

Our agent based model is composed of reactive agents. It is an heterogeneous and closed MAS taking place within a 3D toric space. When brownian movement has been needed, they were approximated by a 3D random walk or equivalent.

Description of agents

In our model, an agent is characterised by its name and its position in a 3D space. The model has mainly two major types of agents: `ReactiveUnitAgent` and `RedoxAgent`. We have used a generic agent `BioAgent` that can handle common attributes (name, weight, position, structure). This class is subclassed into `RedoxAgent` and `ReactiveUnitAgent`.

A `RedoxAgent` is a `BioAgent` which contains a `Body` which is a composite of `MobileParts` and `Joints`. Each `MobilePart` can have a list of `Sites` which in turn have `ReactiveUnitAgent` affinities and redox reaction functions, plus a spatial position. Each site has two states: free or occupied by a `ReactiveUnitAgent`. Sites are used by the `RedoxAgent` to modify the redox value of the `ReactiveUnitAgents` clustered in them.

Model dynamics

During a MAS design, it is necessary to describe the agents life cycle. This life cycle will describe the different rules of interactions between agents and their environment. In our simulations, we have used a generic asynchronous algorithm:

- each agent is randomly inspected,
- its new state is computed using information available through the environment: its positions, its own state and the state of its neighbours.
- its state is updated.

Following the use of HingeFind, FIRST and FRODA algorithms, we have discretized the available spatial positions of the different agents in our system. Using the available movements, we modeled the dynamics and rules controlling them. The rules that control movements depend upon several parameters: random movements, ligand binding/ligand release and redox reactions firing. Simulations combine movements and redox reactions probabilities for the different `ReactiveUnitAgents` at their binding sites.

An application to the complex III of the mitochondrial Respiratory Chain

Complex III plays a central role in Respiratory Chain [CLC+06]. Like all the Respiratory Chain complexes, Complex III is embedded in the inner mitochondrial membrane. It catalyses the following reaction, which oxidizes ubiquinole and reduces cytochrome c:

UQH2 + 2 cytochrome c_{ox} + $2H^+$ = UQ + 2 cytochrome c_{red} + $4H^+$

UQH2 (ubiquinole or Qd) and UQ (ubiquinone) are two different redox states of the same molecule: the CoEnzyme Q.

This reaction process is in fact fairly complex. Mitchell was the first to design the internal work of the complex III (concept in 1967, fully developed by 1975[Mit75]). All the features that occur within the complex III were called the *Q cycle*. At present it is known that during the catalytic process, UQH2 turns into the *semiquinone* redox state ($UQ^{\bullet -}$ or SQ) before reaching the UQ state. Due to the occurrence of the transient free radicals species ($UQ^{\bullet -}$), the respiratory chain is also the site of Reactive Oxygen Species (ROS) production [Cro04]; this is a deleterious process, which can be amplified in mitochondrial diseases and aging.

Structure of the complex III

The Complex III works as a dimer within the mitochondria (see figure 1 where a dimer is represented). One monomer of complex III contains binding sites for UQH2, $UQ^{\bullet -}$, UQ, cytochrome C. Complex III is made of several subunits, among which cytochrome b, the Rieske protein (or Iron Sulfur Protein or ISP) and Cytochrome c1. The cytochrome b contains two fixation sites for UQH2 and UQ (respectively named Qo and Qi). The Rieske protein contains a Fe-S and is close to the cytochrome c1.

The Q Cycle

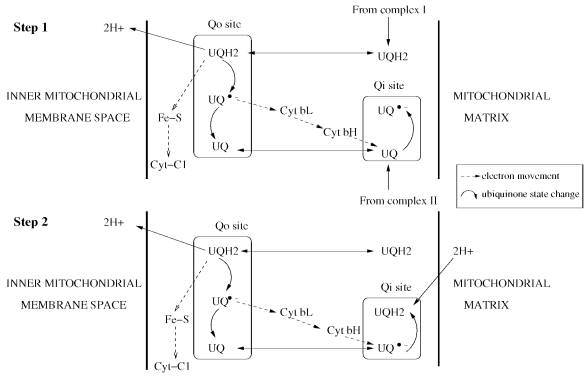


Figure 2: The 2 steps of the Q cycle

A description of the Q cycle is given in figure 2. It consists of two steps.

Step 1: it starts with an electron being transferred from UQH2 (bound to the Qo site) to the *Fe-S* centre of an Iron Sulfur Protein (ISP). This triggers the release of two protons into the inter mitochondrial membrane space and transforms UQH2 into $UQ^{\bullet -}$ (still bound at site Qo). Next, an electron is transferred from the Fe-S to cytochrome c1 and the $UQ^{\bullet -}$ (becoming UQ) gives an electron to the cytochrome b Low potential heme (*cyt bL*). Electron transfer from Low potential heme to High potential heme (*cyt bH*) directly follows. This electron is then given to an UQ molecule at Qi site thus forming $UQ^{\bullet -}$.

Step 2: it starts with the release of UQ from the Qo site. A new UQH2 molecule takes its place. The previously generated $UQ^{\bullet -}$ stays in the Qi site. After that, the scenario is identical to the first step but instead of a UQ, it is a $UQ^{\bullet -}$ that receives the electron from the cyt bH, thus forming UQH2 (on Qi site) with two protons taken from the mitochondrial matrix.
To reach step 2 from step 1 implies that an UQ molecule binds Qi site and that a new UQH2 molecule binds to the Qo site.

Internal movements of the complex III

We have used 18 `Protein Data Bank` structures from bovine complex III to investigate the possible large protein structures movements. Using the HingeFind algorithm, we have divided the complex III dimer into three articulated parts. The first part (central part) corresponds to the cytochrome b, core subunits and the tail of the Rieske proteins. The two other parts correspond to the heads of the Rieske proteins in two conformations linked by an hinge joint.
Within the core part of the dimer, we have used the previously described FIRST and FRODA algorithms to define movements for different binding sites (Qo, Qi) and the iron molecules within the cytochrome c1 and b.

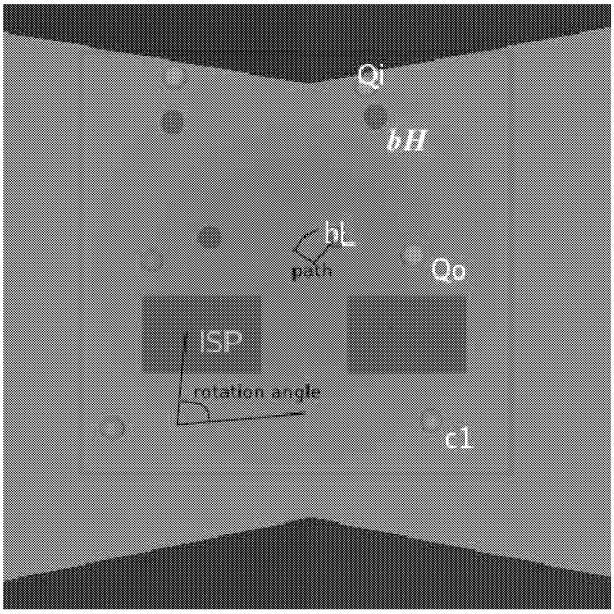


Figure 3: A partitioned complex III

In Figure 3, the different `Sites` of the dimer of complex III as well as the partition and motion of some `Sites` and `MobileParts` are represented.

Resulting Simulations

For our application, a complex III `RedoxAgent` has the following list of `Sites`: `cyt bL`, `cyt bH`, `ISP`, `cyt c1`, `Qo` and `Qi` that can contains `ReactiveUnitAgents`: an iron (cyt bL, cyt bH, ISP), a cytochrome c (cyt c1) or a quinone (Qo, Qi).
For calibrating and extending the set of rules for movements, we have used previously published parameters given in [ICBS99] [EGY+06] [CLC+06]. In a similar way, the redox reactions modeling used a set of initial values common to previous modeling attempt [DGKW01] and refined experimental data [DM94].
We developed a prototype tool to automatize `HingeFind` and `FIRST/FRODA` analysis (called 'Foral') and a prototype simulator for our agent model (called 'AMORSS'). These prototypes are both GPL licensed softwares available on request. Among many simulations that explored the parameters space, the figure 4 shows results obtained using a total of 67 CoEnzyme Q and 20 cytochrome c1 `ReactiveUnitAgents`, with one Complex III `RedoxAgent`, within a ten million angstrom square space, with 10^{-3} second of simulated time per simulation step. In figure 4, one step represents 100 simulated time steps. In a common nowadays desktop, it takes 5 to 10 minutes to generate nearly 10 000 time steps.

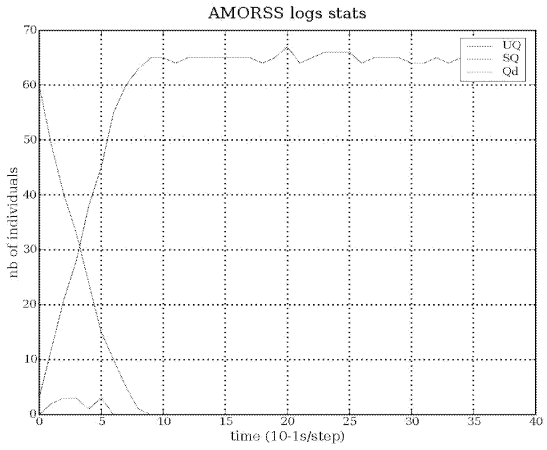


Figure 4: Complex III overall reaction kinetic

On top of the kinetics, we can follow the individual path of the different electrons that are exchanged within the system. We were able to simulate the non pathological and canonical functioning of the Q cycle. We were also able to see many other electron paths that can lead to biologically relevant situations. A simple example is that during simulations, some reactions go back and forth more than three quarter of the time. But they do not significantly slow the overall biological system, mainly because they are the fastest reactions in the system and the overall speed is heavily controlled by the slowest reactions. The loss of energy due to the fact of some reactions go back and forth can be of concern in certain pathologies. We are currently analysing in throughly the different biological meaning of the uncovered electron paths.

Conclusion and perspectives

To conclude, we have developed a generic method to assess redox enzyme functioning by integrating geometric partitioning algorithms and agent based simulation. We are able to find the Q-cycle functioning of the com-

plex III overall reaction as a consequence of the reaction mechanisms taking into account the tridimensional structure of the complex and its conformational changes. We are now able to access the possible short-circuits and deleterious electron transfer mechanisms within the Respiratory Chain enzymatic complexes. We plan on analysing as much as possible the current results on the complex III and then apply our model to simulate the whole Mitochondrial Respiratory Chain at work.

REFERENCES

[BKW+77] F C Bernstein, T F Koetzle, G J Williams, E F Jr Meyer, M D Brice, J R Rodgers, O Kennard, T Shimanouchi, and M Tasumi. The Protein Data Bank. A computer-based archival file for macromolecular structures. *Eur J Biochem*, 80(2):319–324, Nov 1977. Duplicate Publication.

[CLC+06] Antony R Crofts, Sangmoon Lhee, Stephanie B Crofts, Jerry Cheng, and Stuart Rose. Proton pumping in the bc1 complex: a new gating mechanism that prevents short circuits. *Biochim Biophys Acta*, 1757(8):1019–1034, Aug 2006.

[Cro04] Antony R Crofts. The cytochrome bc1 complex: function in the context of structure. *Annu Rev Physiol*, 66:689–733, 2004.

[DGKW01] O V Demin, I I Gorianin, B N Kholodenko, and H V Westerhoff. [Kinetic modeling of energy metabolism and generation of active forms of oxygen in hepatocyte mitochondria]. *Mol Biol (Mosk)*, 35(6):1095–1104, Nov 2001.

[DM94] P L Dutton and C C Mosser. Quantum biomechanics of long-range electron transfer in protein: hydrogen bonds and reorganization energies. *Proc Natl Acad Sci U S A*, 91(22):10247–10250, Oct 1994. Comment.

[DM95] Jacobs DJ and Thorpe MF. Generic rigidity percolation: The pebble game. *Phys Rev Lett*, 75(22):4051–4054, Nov 1995. JOURNAL ARTICLE.

[EGY+06] Lothar Esser, Xing Gong, Shaoqing Yang, Linda Yu, Chang-An Yu, and Di Xia. Surface-modulated motion switch: capture and release of iron-sulfur protein in the cytochrome bc1 complex. *Proc Natl Acad Sci U S A*, 103(35):13045–13050, Aug 2006.

[ICBS99] S Izrailev, A R Crofts, E A Berry, and K Schulten. Steered molecular dynamics simulation of the Rieske subunit motion in the cytochrome bc(1) complex. *Biophys J*, 77(4):1753–1768, Oct 1999.

[JRKT01] D J Jacobs, A J Rader, L A Kuhn, and M F Thorpe. Protein flexibility predictions using graph theory. *Proteins*, 44(2):150–165, Aug 2001.

[KMM+03] Salim Khan, Ravi Makkena, Foster McGeary, Keith Decker, William Gillis, and Carl Schmidt. A multi-agent system for the quantitative simulation of biological networks. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 385–392, New York, NY, USA, 2003. ACM Press.

[Mit75] P Mitchell. The protonmotive Q cycle: a general formulation. *FEBS Lett*, 59(2):137–139, Nov 1975.

[PB04] A. Zemirline et L. Marc P. Ballet. The biodyn language and simulator. application to an immune response and e.coli and phage interaction. *Journal of Biological Physics and Chemistry 4 (2004) 93-101*, September 2004.

[SMW95] R A Sayle and E J Milner-White. RASMOL: biomolecular graphics for all. *Trends Biochem Sci*, 20(9):374, Sep 1995.

[SWT05] B.M. Hespenheide S. Wells, S. Menor and M.F. Thorpe. Constrained geometric simulation of the diffusive motions in proteins. *Phys. Bio.*, 2:127–136, 2005.

[WS97] W Wriggers and K Schulten. Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. *Proteins*, 29(1):1–14, Sep 1997.

[WW06] Ken Webb and Tony White. Cell modeling with reusable agent-based formalisms. *Applied Intelligence)*, 24(02):169–181, 2006.

VERIFICATION OF POTENTIAL BINDING SITES OBTAINED WITH STOCHASTIC ROADMAP SIMULATION

Marcin Pacholczyk, Zbigniew Starosolski
Silesian University of Technology
Institute of Automatic Control
Akademicka 16
44-100 Gliwice
POLAND
marcin.pacholczyk@polsl.pl zbigniew.starosolski@polsl.pl

KEYWORDS
Molecular Docking, Stochastic Roadmap Simulation, Interaction Surfaces, Poisson-Boltzmann Equation, LUDI.

ABSTRACT

In the paper we present a method of visual verification of potential binding sites obtained with stochastic roadmap simulation applied to the problem of molecular docking. Molecular docking can be regarded as a potential method for computer aided drug design and optimization. Inspired by probabilistic path planning in robotics, stochastic roadmap methodology can be regarded as a very interesting approach to effective sampling of ligand conformational space around a protein molecule. Protein - ligand interaction in divided into two parts electrostatics, modeled by the Poisson-Boltzmann equation, and Van der Waals interaction represented by the Liennard-Jones potential. The stochastic roadmap combined with geometrical model of protein – ligand interaction, such as LUDI, could give full insight into the molecular docking problem from unbound phase to the final phase when a ligand is bound to the active site of a protein.

INTRODUCTION

Shortly speaking the molecular docking can be formulated as a task of fitting ligand (drug molecule) into an active (also called binding) site of a receptor (protein) and can be regarded as a potential solution to virtual drug design problem. Successful docking process requires ligand to match geometrically and energetically the protein binding site. Many approaches has been developed to solve the problem of molecular docking (Perola et al. 2004). However, to the best of our knowledge all of them, due to the high complexity of the problem, consider the specific aspect of the molecular docking process only, and as such do not capture the whole dynamics of the process from the beginning – the unbound state (which I would call the transient state using analogy from control theory), when the ligand is loose, away from its binding site, being influenced by a force field generated by the protein and aquatic solution, to the end – the bound state (the steady state), when the ligand is being kept in the binding pocket of the protein by binding site specific interactions like hydrogen bonds or hydrophobic contacts.

Molecular Docking as a Static Process

Most of the molecular docking algorithms were developed to be useful in virtual High Throughput Screening (vHTS) experiments, named so after the real laboratory process – screening of thousands of potential lead structures for new drugs development. That approach is not very accurate in modeling protein-ligand interactions but such methods are fast enough to screen an entire, large drug candidates library in reasonable time (single minutes per one compound), successfully discovering new lead structures. Usually vHTS methods incorporate some general protein-ligand interaction model form which a proper scoring function is derived to estimate the affinity of binding. The view of the molecular docking process is often limited to searching the best possible pose (including rigid body transformations as well as ligand internal flexibility, usually limited to its dihedral angles) of the ligand within the binding site the location and local configuration of which is presumed known. The protein is assumed rigid and immobilized. The searching of the best possible pose of the ligand in the binding pocket of the protein is usually conducted by a search algorithm trying to satisfy the requirements of chosen scoring model (maximization or minimization of the scoring terms). Although, such methodology does not answer to the problem if it is possible for the ligand to enter the binding site in particular (best scoring) conformation or where is the binding site located (we must know the location a priori), incorporated with a proper model such as LUDI (Böhm 1992), can give us an interesting insight of what is really going on in the binding site. We are able to detect various kinds of crucial interactions between the ligand and its binding site and what is the most interesting to localize them in 3D space, what gives us qualitative and quantitative characteristic of the particular binding site. We can for example show which particular atoms of the ligand form hydrogen bonds with particular atoms making protein amino acids or which part of a ligand is influenced by hydrophobic or other types of interactions. Of course those methods are not yet perfect and there is still a lot of work to do concerning search algorithms and ranking the docked structures.

The Dynamics of Molecular Docking

On the other hand there are a few methods, being rather theoretical in nature, trying to directly account for protein-ligand interaction dynamics. The classical molecular

dynamics (MD) simulation is regarded to be impractical in this case, due to its extremely high computational demand. Only a few nanoseconds of such simulation is within the range of todays super computers. However, there are methods offering fair approximation of the protein-ligand interaction dynamics, based on effective sampling of the combined transformational and conformational space of the ligand. One of the most interesting Monte Carlo (MC) type methods is the Stochastic Roadmap Simulation (SRS) (Apaydin 2004; Apaydin et al. 2003; Apaydin et al. 2002; Singh et al. 1999) a method derived from robotic motion planning methodology. Unlike classical MC approach, SRS enables one to sample from all possible paths the ligand may choose entering the binding site. The basic idea is to effectively guess some intermediate configurations of the ligand and to obtain a distribution of energetically favorable paths to the binding site (through these intermediate configurations). The original SRS approach is capable of detecting putative binding sites or even distinguishing the catalytic binding site (Apaydin et al. propose time to escape from so called funnel of attraction as a measure of binding affinity) (Apaydin 2004; Apaydin et al. 2003; Apaydin et al. 2002). However, SRS does not give any information about nature of interactions between the ligand and the binding site (hydrogen bonds etc.), and should be combined with one of the previously mentioned static models in order to analyze the bound (steady) state. We propose to use LUDI model (Böhm 1992) as a tool of visual inspection of qualitative and quantitative properties of the interaction between the binding site of a protein and the bound ligand.

MODELS OF BIOMOLECULES

Modeling the Ligand (SRS Method)

By ligand we understand small molecule with small number of degrees of freedom (max. 10-15). First we assign to one terminal atom (a base) three cartesian coordinates (x, y, z) which describe the location of the ligand in space and two angles (α, β) describing orientation of base bond, second we assign one dihedral angle for each non-terminal atom (conformational degrees of freedom) see Fig. 1. Structures of rings are assumed constant. Internal interactions between atoms are divided into Coulombic and van der Waals interaction (Apaydin et al. 2002)

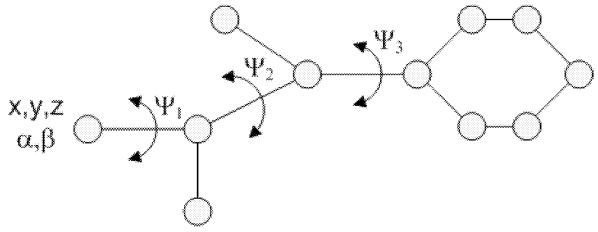


Figure 1: Degrees of freedom of a hypothetical ligand

Force Field Around Protein (SRS Method)

The environment around a protein molecule is modeled by electrostatic and van der Waals interactions. These interactions generate attractive and repulsive forces which cause motion of molecules (Apaydin et al. 2002). The electrostatic part is modeled by Poisson-Boltzmann equation (PBE) (Sharp and Honig 1990).

$$\nabla\left[\varepsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r})\right]-\varepsilon(\boldsymbol{r})\kappa^2\sinh\left[\phi(\boldsymbol{r})\right]+4\pi\rho(\boldsymbol{r})=0 \quad (1)$$

where: ε -dielectric constant, ϕ - electrostatic potential, ρ - charge density, κ - inverse Debye-Hueckel length, $\boldsymbol{r}$ – location vector in 3D space.

The model associated with PBE (1) is far more accurate in this case than simple Coulombic models and incorporates features such as location dependent dielectric constant (as dielectric constant varies in space – experimentally determined dielectric constant for proteins equals approx. 2 for water equals 80) (Sharp and Honig 1990) and mobile ions contribution to the electrostatic potential (natural environment for proteins is usually salty aquatic solution). Protein is considered a rigid body limited by solvent accessible surface (Conolly 1983). Analytical solution to the PBE is possible only for systems with very simple geometries (protein surface contains many clefts and cavities) thus we use numerical solution obtained using finite difference methods. In order to solve PBE on 3D grid we use two computer programs DelPhi (Rocchia et al. 2001) and APBS (Holst and Saied 1995). The software is highly configurable concerning simulation parameters as well as PBE solver parameters. Additionally APBS has an excellent companion visualization software VMD (Humphrey et al. 1996). Van der Walls interactions are derived using Lennard-Jones potential (2) calculated for the same 3D grid as for PBE.

$$E_v = \in_0 \left[\left(\frac{r}{r_0}\right)^{12} - 2\left(\frac{r}{r_0}\right)^{6}\right] \quad (2)$$

where: r – distance between atoms (Euclidean), $r_0, \in_0$ – material dependent constants

Interaction Surfaces (LUDI Model of Protein-Ligand Interactions)

As a model for protein-ligand interaction we use LUDI model first introduced by Böhm (Böhm 1992) and adopted to the form of interaction surfaces by Rarey *et al.* (Rarey et al. 1996). The model considers several possible types of intramolecular interactions i.e. hydrogen bonds or hydrophobic contacts. The interaction itself is modeled by an interaction center and interaction surface located on a sphere with center in interaction center. The interaction to take place requires interaction center of the ligand to lie on interaction surface of the protein and vice-versa. An interaction surface is modelled as a discrete set of points in 3D space. Such representation has an advantage of encoding a set of biochemical rules governing intramolecular interactions in compact geometric form easy to store and process. See Fig. 2 for a few examples of molecular interaction surfaces.The data concerning other types of interaction surfaces can be found in (Böhm 1992; Rarey et al. 1996). We use our own implementation of LUDI and interaction surfaces in Matlab 7.

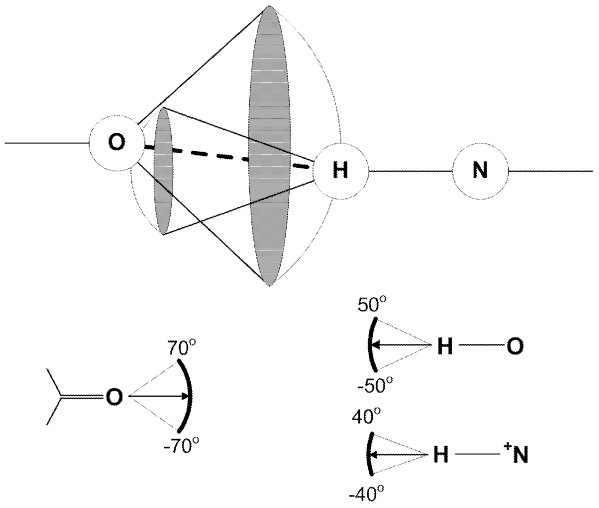


Figure 2: Illustration of the idea of modeling molecular interactions by interaction surfaces. Examples of interaction surfaces. An interaction takes place if an interaction center of a ligand lies on interaction surface of a receptor and vice-versa(upper part). Oxygen atom (lower left) as a acceptor and oxygen and nitrogen (lower right) as donors of hydrogen bond. Interaction surfaces are spherical sections with 1.9 Å radius.

STOCHASTIC ROADMAP SIMULATION

The Stochastic Roadmap Simulation (SRS) (Apaydin 2004; Apaydin et al. 2003; Apaydin et al. 2002) is a method for effective sampling ligand conformational space around protein. The SRS was inspired by probabilistic motion planning in robotics (Singh et al. 1999).

Roadmap Construction

Apaydin *et al.* (Apaydin et al. 2002) define a roadmap as a discrete representation of molecular motion. Each node of a roadmap represents one conformation of a ligand. Formally each conformation of n parameters is represented by a vector $[q_1, q_2, ..., q_n]$. The set of all possible conformations forms the conformation space C. SRS assumes that the interactions are described by an energy function $E(\boldsymbol{q})$, which depends only on the conformation $\boldsymbol{q}$ of the ligand. A pathway in C represents motion of the ligand around protein. A roadmap is a directed graph G encoding many pathways in C. Each node of a roadmap is randomly selected conformation $\boldsymbol{q}$ from C with associated energy $E(\boldsymbol{q})$. Each directed edge between two nodes v_i and v_j have associated weight Pij which is a probability of transition between the two nodes. In order to construct a roadmap the algorithm must sample n conformations randomly and independently from C. Then for each node v_i we must find k nearest neighbors of v_i according to selected metric (i.e. RMSD or Euclidean). After that a transition probability Pij is computed for every pair of neighboring nodes. Pij calculation is based on difference of energy E(q) between nodes vi and vj according to the formula:

$$P_{ij} = \begin{cases} 1/N_i \exp(-\Delta E_{ij} / k_B T) & if \quad \Delta E_{ij} > 0 \\ 1/N_i & otherwise \end{cases} \tag{3}$$

where k_B – Boltzmann constant, T – system temperature, N_i – number of neighbors of node v_i. The self transition probability is defined as:

$$P_{ii} = 1 - \sum_{j \neq i} P_{ij} \tag{4}$$

which ensures that the transition probabilities from any node sum up to 1 (Apaydin et al. 2002).

Simulation

Although it is possible to perform a simulation on a roadmap which corresponds to a discrete version of standard Monte Carlo method (discretization is defined by a roadmap) Apaydin *et al.* (Apaydin et al. 2002) suggest that we never need to generate individual trajectories on a roadmap but rather calculate for the parameter of interest. As an example time to escape (in simulation steps) from funnel of attraction around protein binding site is given. Apaydin et al. propose the escape time as a measure of affinity of a ligand to a putative binding site. The funnel of attraction F_i is defined as conformations within 10 Å of the bound conformation. Time to escape can be easily calculated using first step analysis from Markov chain theory by solving the following system of equations (Apaydin et al. 2002):

$$\tau_i = 1 + \sum_{v_j \in F_i} P_{ij} \cdot \tau_j \tag{5}$$

where τ_i– time to escape starting from i-th node, F_i - funnel of attraction around i-th binding site, v_i – i-th node.

Predicting the Putative Binding Sites

According to Apaydin et al. (Apaydin et al. 2002) the SRS method is capable of predicting putative binding sites of a protein. The process consists of several consecutive steps. First a large number (i.e. 10000) of nodes is sampled at random. Next the resulting nodes are ordered by increasing energy, then a small part of lowest energy nodes (i.e. 10) is selected for the next step which is sampling of additional nodes in the direct neighborhood of previously selected nodes. Finally the resulting lowest energy nodes are filtered by pairwise distance of 10 Å from each other (in order to avoid multiple representations of the same putative binding site region) and by 5 Å distance from the protein surface (a binding site should be found close to the protein surface).

RESULTS AND DISCUSSION

As a model system we used crystal structure of dogfish M4 apo-lactate dehydrogenase (PDB ID: 1ldm) (Abad-Zapatero et al. 1987; Berman et al. 1992) and its ligand oxamate downloaded from the PDB database (Berman et al. 1992). We run SRS a 1000 times and obtained several estimations of potential binding site locations. Detailed description of the process leading to finding putative binding sites can be found in the section Predicting the Putative Binding Sites.

We present results obtained for the catalytic site and three selected putative binding sites (lowest energy ligand conformations). The catalytic site is the most important region of an enzyme where it performs its catalytic function, in this case oxidoreduction of lactate to pyruvate. Interactions between the protein and the ligand in the catalytic site are usually revealed by crystal structure deposited in the PDB database. Our goal was to verify that the LUDI model is capable of detecting important interactions between the ligand and the catalytic site of the protein and to check whether there exist significant protein-ligand interactions in the putative binding sites predicted by the SRS method and therefore to estimate the binding affinity of the ligand to the putative binding site.

For the catalytic binding site we were able to detect hydrogen bonds between oxamate and ARG106 and ARG169 residues of the lactate dehydrogenase (see Fig.3). However, for the putative binding sites we detected a hydrogen bond only in one (see Fig.4) out of three cases (see Fig.4-6).

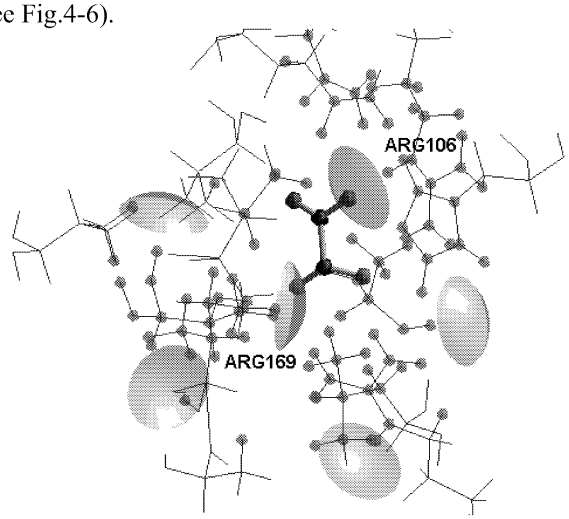


Figure 3: The catalytic site of dogfish M4 apo-lactate dehydrogenase. The model reveals two hydrogen bonds between oxamate and ARG106 and ARG169 residues of the protein. Hydrogen bond donors are shown as spheres.

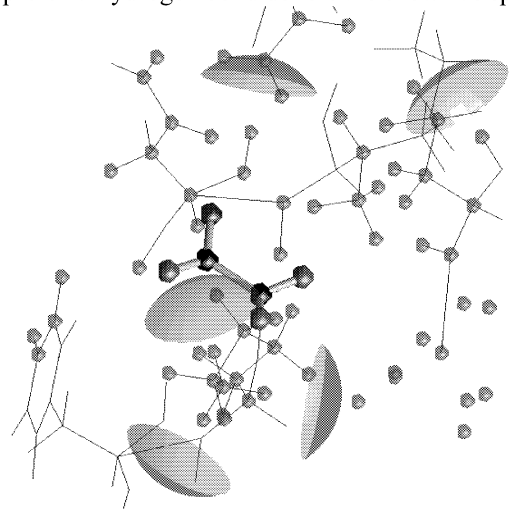


Figure 4: Putative binding site 1. One hydrogen bond to the protein structure was found. Hydrogen bond donors are shown as spheres.

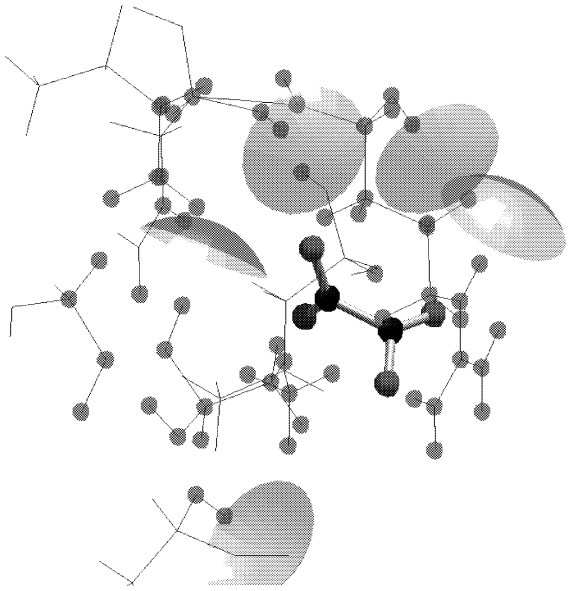


Figure 5: Putative binding site 2. No direct protein-ligand interactions were found. Hydrogen bond donors are shown as spheres.

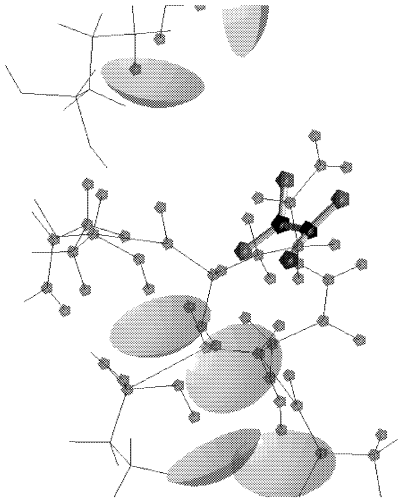


Figure 6: Putative binding site 3. No direct protein-ligand interactions were found. Hydrogen bond donors are shown as spheres.

CONCLUSIONS

In this paper we presented an application of a statistically derived geometrical model of protein-ligand interactions to the characterization of putative binding sites obtained with the SRS method. Clearly our method is capable of detecting significant protein-ligand interactions (like hydrogen bonds) what has been shown for the catalytic site. However, in case of putative binding sites in two out of three cases we were unable to detect any direct protein-ligand interactions even though there exist potential hydrogen bond donors in all the binding sites. We suggest that a kind of optimization of the placement of the ligand inside putative binding site (whether energy or geometry based, i.e. pose clustering proposed in (Rarey et al. 1996)) would improve the ligand binding affinity and probably reveal additional protein-ligand interactions. We conclude that our approach has a great potential of characterization of protein-ligand

interactions however more research is needed concerning another protein examples and different binding sites configurations.

ACKNOWLEDGEMENTS

The work has been sponsored by the Polish Government funds for science in the years 2007-2008 as a research project. The authors would like to express their gratitude for invaluable support from prof. Marek Kimmel (Rice University / Silesian University of Technology).

REFERENCES

Abad-Zapatero, C., Griffith, J.P., Sussman, J.L., Rossmann, M.G. 1987. "Refined crystal structure of dogfish M4 apo-lactate dehydrogenase". *Journal of Molecular Biology*, 198, 445-67.

Apaydin M. S., Brutlag D. L., Guestrin C. E., Hsu D., Latombe J-C., Varma C. 2003. "Stochastic Roadmap Simulation: An Efficient Representation and Algorithm for Analyzing Molecular Motion". *Journal of Computational Biology*, 10 (3-4), 257-281.

Apaydin M. S., Guestrin C. E., Varma C., Brutlag D. L., Latombe J-C. 2002. "Stochastic roadmap simulation for the study of ligand-protein interactions". *Bioinformatics*, 18 sup. 2, S18-S26.

Apaydin M.S. , 2004. "Stochastic Roadmap Simulation: An Efficient Representation and Algorithm for Analyzing Molecular Motion". *Ph. D. thesis, Stanford University.*

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. 1992. "The Protein Data Bank". *Nucleic Acids Research*, 28, 235-242.

Böhm, H-J. 1992. "LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads". *Journal of Computer Aided Molecular Design*, 6, 593-606.

Conolly M.L. 1983. "Solvent-accesible surfaces of proteins and nucleic acids". *Science* 221(4612) 709-713.

Holst M. J., Saied F. 1995. "Numerical solution of the nonlinear Poisson-Boltzmann Equation: Developing more robust and efficient methods". *Journal of Computational Chemistry*, 16 (3), 337-364.

Humphrey, W., Dalke, A. and Schulten, K. 1996. "VMD - Visual Molecular Dynamics", *Journal of Molecular Graphics*, , vol. 14, 33-38.

Perola E., Walters W.P., and Charifson P.S. 2004. "A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance". *Proteins*, 56, 235-249.

Rarey, M., Wefing, S., Langauer, T. 1996. "Placement of medium-sized molecular fragments into active sites of proteins". *Journal of Computer Aided Molecular Design*, 10, 41-54.

Rocchia W., Alexov E., Honig B. 2001. "Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions" *Journal of Physical Chemistry B*, 105, 6507-6514.

Sharp K.A., Honig B. 1990. "Electrostatic interactions in macromolecules: Theory and applications". *Annual Reviews of Biophysics and Biophysical Chemistry*, 19, 301-332.

Singh, A.P., Latombe, J.C., and Brutlag, D.L. 1999. "A motion planning approach to flexible ligand binding". In *Proceedings of the 1999 International Conference on Intelligent Systems for Molecular Biology* (Heidelberg, Germany, Aug. 6-10), 252–261.

AUTHOR BIOGRAPHY

MARCIN PACHOLCZYK was born in Bytom, Poland and went to the Silesian University of Technology in Gliwice, where he studied Automatic Control and Robotics. He obtained his M. Sc. degree (Bioinformatics) in 2003, and joined the Control and Robotics Group. Since 2003 he has been doctoral student at the Institute of Automatic Control, working on application of Computer Vision and Robotics methods to the problem of Molecular Docking. His research interests are: Molecular Modeling and Simulation and Bioinformatics.

ZBIGNIEW STAROSOLSKI was born in Bytom, Poland and went to the Silesian University of Technology in Gliwice, where he studied Automatic Control and Robotics. He obtained his M. Sc. degree in 1999, and joined the Control Theory Group. In 2004 he obtained his Doctor's Degree and joined Control and Robotics Group as an Assistant Professor. Since 2007 he has been working in Systems Engineering Group and teaching Biostatistics and Biometry. His scientific interests are: Control Theory, Molecular Biology and Biostatistics.

FOREST FIRE SPREAD MODELLING: A PRACTICAL GIS-BASED APPROACH

Yves Dumond

Laboratoire LISTIC, Université de Savoie
Campus Scientifique
F-73376 Le Bourget-du-Lac Cedex
E-mail : Yves.Dumond@univ-savoie.fr

KEYWORDS
Forest fire spread modelling, forest fire suppression, geographic information system, tactical situation, cellular automata.

ABSTRACT

We report in this paper on an experiment in implementing a software system dedicated to forest fire simulation. This is part and parcel of an integrated forest fire suppression management system called Asphodèle. First, we describe the different input data required by our model, i.e. the landscape classification, the altimetric database and the set of wind matrices. We also introduce a formula that provides a fire's basic rate of spread using a given set of meteorological data. Then, we show how this rate of spread can be combined with the aforementioned data in order to get local rates of spread, the effect of which is managed by means of a cellular automata based approach.

INTRODUCTION

Every year, forest fires devastate several hundred thousands hectares of land in the European Union and in many other parts of the world. Though they are sometimes regarded as a natural phenomena that regenerate woody areas, forest fires are generally ecological disasters that lead to large social and economic damage. Furthermore, in Europe, because of the high population density, inhabited areas are often threatened.

Therefore, much research work, that ranges from potential danger evaluation techniques to fire suppression strategies, is dedicated to this specific curse. Hence, the modelling and simulation of the propagation of forest fires is a key issue. As a matter of fact, this challenging field of research can be a great help in forest fire suppression management.

FRAMEWORK OF THE PROJECT

The work presented in this paper is part of the Asphodèle project (Dumond 2006). Since 2002, the University of Savoie and a fire brigade in the South of France have entered into a partnership that aims to provide a global software system for forest fire suppression management. This system is supported by a geographical information system (GIS). The following lists the main functions carried out by the Asphodèle system:

1. The management of numerical battlefields called "tactical situations": these are specific diagrams (cp. Figure 1). Their elaboration consists in adding dedicated graphical items to background maps. The latter can as well be geological survey maps or aerial photographs. Basically, they include many structures built by man, e.g. roads, buildings, high voltage lines, fire hydrants, etc. On this basis, the officer in charge of the tactical situation management captures the information regarding the fire, e.g. the starting point, the related areas and the different spreading axes. Fixed means, e.g. water resources, hospitals, heliports, filling stations, etc. are also specified. At the end, mobile means, e.g. tankers, oil supply vehicles, logistics, mobile headquarters, medical units, etc. are captured as well as the actions they perform, e.g. flank attacks, support lines that aim at stopping the fire, etc.

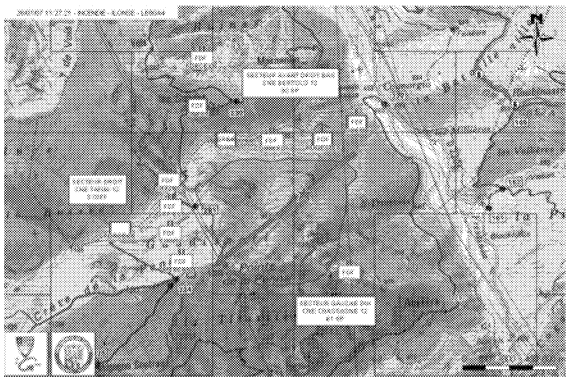


Figure 1: A tactical situation © SDIS06

2. The management of the different material and human mobile means involved in the fight: a specific software component, called "mobile resources manager" (MRM), provides the situation at all times of each firefighting unit from the initial requirement to the time it is finally relieved. The coherence of the data shared by the current tactical situation and the MRM, e.g. the location of the different units engaged on the ground, is automatically undertaken by the system.

3. A decision-making support module that helps implement a prospective step which is a cornerstone in the management of the interventions. This task is entrusted to experienced officers who are able to

predict fire growth according to the various available data (cp. Figure 2).



Figure 2: Fire suppression management with the Asphodèle software system © JG Bouillon / SDIS06

These informal reasonings are in particular based on a 3-D display of the operating site and on careful study of the aerial photographs. In particular, these analyses have to take into account the nature of the vegetation and its density. Thus, it turns out that the predicted spread of the fire is part and parcel of the set of tasks that must be handled by the command line. This is the reason why we intend to implement an automatic tool for the purpose. In fact, our objective is to be able to compare the results of simulations to those obtained by human-based reasoning.

Let us also note that, because it is fully integrated in the global command-line, fire spread predictions will generally be subject to severe timing constraints. In addition, we have to cope with the fact that all the Asphodèle software components, including the simulator module, must run on portable computers and consequently that they should not require huge computing ressources.

FOREST FIRE SPREAD MODELLING

Models

Forest fire spread modelling has been the subject of many kinds of work, especially in North America. For two interesting syntheses about this subject, see (Weise and Biging 1997) and (Perry 1998). These works can be divided into three groups:

1. Theoretical works which focus on analytical descriptions of the phenomena through physical models. As they stand, these models are not suitable for application.

2. Empirical studies are based on the gathering of data related to actual forest fires. A typical example of such an approach is the Canadian Forest Fire Behaviour Prediction System (Forestry Canada Fire Danger Group 1992) which is a synthesis of thousands of observations carried out in the field since the 1920s. Other works include (Martins Fernandes 2001) and (Dimitrakopoulos and Drita 2006). However, the main drawback of empirical steps is that their results cannot be applied to frameworks different from the one in which they have been elaborated.

3. Semi-empirical approaches mix the two previous ones. A well known model of this kind is due to Rothermel (Rothermel 1972). Two widely used tools, i.e. FARSITE (Finney 1998) and BEHAVE (Rabner et alt. 2001), stem from this model.

The proposed approach

According to the works referenced above, wind (Nelson 2002), slope and vegetation (in particular its moisture content), are the main variables that affect the rate of spread of forest fires. The issue of the respective influence of wind and slope has been widely discussed (Weise and Biging 1997), (Viegas 2004). However, the specific framework of this work, i.e. important forest fires aided by high velocity winds, leads us to lay down as a principle the fact that wind is the predominant variable. Moreover, it is worth noting that the work presented in this paper has many common points with spread modelling using cellular automata (Richards 1994), (Bodrozic and alt. 2006), for instance 2-D landscape representation by means of a set of cells.

INPUT DATA

Lanscape classification

As stated above, Asphodèle is a GIS-based software system. Therefore, we have a set of geological survey maps and aerial photographs available under different scales. This is enough to classify the potential operating zones according to their presumed flammability.

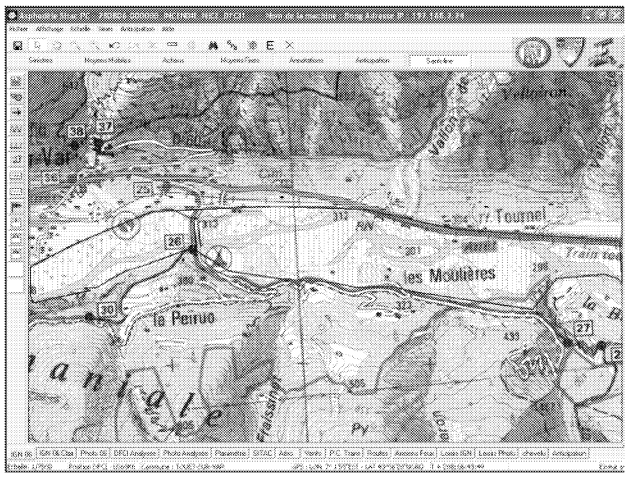


Figure 3: Example of areas classification © SDIS06

Thus, a given area can either be *non-flammable*, *low-flammable* or simply *flammable*. The latter is the default option, i.e. it does not require any explicit specification of the corresponding zones. For instance, in Figure 3, the river's bed (in blue) has been classified as a non-flammable area and the band of land right above (in green) as a low-flammable area while the rest of the field is regarded as flammable.

The altimetric database

The altimetric database consists in a 2-D rectangular matrix representing lanscape. The dimensions of the cells are 50 m x 50 m. Thus, the altitude of the geographic center of each of them is provided.

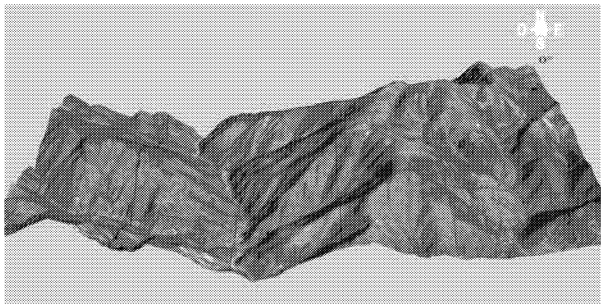


Figure 4: 3-D Visualization © SDIS06

In combination with maps or aerial photographs, this database allows the display of 3-D views of the lanscape as shown in Figure 4.

The wind matrices

Following the example of the altimetric database, the wind matrices are landscape grids made of cells, the dimensions of which are 150 m x 150 m. Moreover, each cell belonging to a wind matrix contains exactly nine cells of the altimetric database insofar as the two kinds of grids can be exactly superposed.

However, on the basis of a global direction and of a fixed wind velocity, each cell contains a local direction and a wind speed due to the relief (cp. Figure 5).

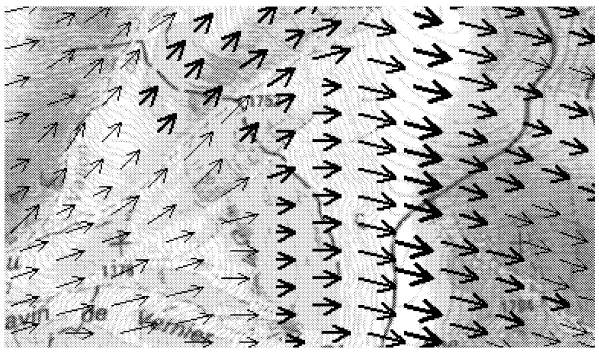


Figure 5: Part of a wind matrix © SDIS06

The wind speed is supposed to be evaluated at 10 m high. The set of matrices (we have a specific matrix for each dominant wind) has been purchased from a private firm specialized in the simulation of fluid flow. Each matrix has been computed with a global speed of 10 $m.s^{-1}$. However, the results of the simulation remain relevant for speeds ranging from 7 to 20 $m.s^{-1}$. Therefore, starting from a given wind velocity between 7 and 20 $m.s^{-1}$ and different from 10 $m.s^{-1}$, it is possible to deduce the actual local speed from the values present in the matrix by means of a simple rule of three. Thus, each wind matrix may appear as a kind of abacus. Let us also note that the values belonging to the aforesaid interval (i.e. 7 to 20 $m.s^{-1}$), must be regarded as *average* speeds and not maximum values reached by winds blowing in gusts.

THEORETICAL BACKGROUND

The effect of meteorological conditions

The Thornthwaite formula (Thornthwaite 1948) provides an estimation (on flat land) of the fire's rate of spread R_s in function of the temperature in the shade, the soil water supply and the speed velocity:

$$R_s = 180*\exp(0.06*T_a)*\mathrm{tgh}((100-S_w)/150) *(1 + 2*(0.8483 + \mathrm{tgh}(W_v/30 - 1.25))) \quad (1)$$

where :

1. R_s: rate of spread (in $m.s^{-1}$).
2. T_a: temperature in the shade (in degrees celsius).
3. S_w: soil water supply (this parameter ranges from 0 to 150). Whenever $100 \leq S_w$ no ignition is possible and if $S_w \leq 50$ the vegetation is in state of dryness. Furthermore, if $S_w \leq 30$, the vegetation is in a state of hydric stress, i.e. it may die because of lack of water.
4. W_v: wind velocity (in $km.h^{-1}$).
5. exp: exponential function.
6. tgh: hyperbolic tangent function.

Though it is a bit old, this formula provides reliable spread predictions. Furthermore, the firemen involved in the project have acquired, year after year, great expertise in the interpretation of the results that it provides.

Slope effect

We now turn to the effect of the slope on fire spread. Insofar as we have admitted that the variables which appear in the Thornthwaite formula, especially the wind speed, were the dominant factors, slope effect is expressed in our model by means of numerical coefficients. Thus, starting from a base rate on flat land R_0, the rate of spread due to the slope R is given by:

$$R = C(A)*R_0 \quad (2)$$

where:

1. R: corrected rate of spread.

2. C(A): coefficient defined in function of the angle of the slope.
3. R_0: base rate of spread.

It is well known that a fire burning uphill aided by wind has the highest rate of spread. Indeed, this rate increases exponentially with the slope. Therefore, many works have been dedicated to this specific issue, e.g. (Nelson 2002) and (Viegas 2004). In (Cheney 1981), the coefficient which must be applied in that case is provided by an equation in the form:

$$C_{uh}(A) = \exp(c_0 * A) \qquad (3)$$

where:

1. $C_{uh}(A)$: coefficient for uphill spread.
2. A: angle of the slope with the horizontal axis (positive value in degree).
3. c_0: parameter whose value is equal to 0.0693 in (Cheney 1981).

There are very few works concerning the spread of forest fires downhill. As a matter of fact, (Van Wagner 1988) is likely to be the sole paper published on this subject. According to this work, the coefficient which must be applied in case of downhill spread is given by an equation in the form:

$$C_{dh}(A) = c_0 - c_1 * A + c_2 * A^2 \qquad (4)$$

where:

1. $C_{dh}(A)$: coefficient for downhill spread.
2. A: angle of the slope with the horizontal axis (positive value in degree).
3. c_0, c_1 and c_2: parameters whose repective values are equal to 1, 0.330 and 0.000749 in (Van Wagner 1988).

THE GLOBAL PROCEDURE

The global procedure main data is a 2-D array of elements, hereafter denoted LD. The structure of these elements is the following:

```
Type Element:
        Struct
            Geographical_coordinates : (GPS_x, GPS_y);
            Altitude: N;
            Flammability: (non-flammable,
                           low-flammable, flammable);
        End_struc;
```

In fact, each element of this array corresponds to an element of the altimetric database added to an information related to the local level of flammability. The latter clearly depends on the local vegetation. On the basis of the different concepts indicated above, the list of the tasks successively performed by the global procedure is as follows:

1. The initial fire area is drawn on the map (cp. Figure 6).

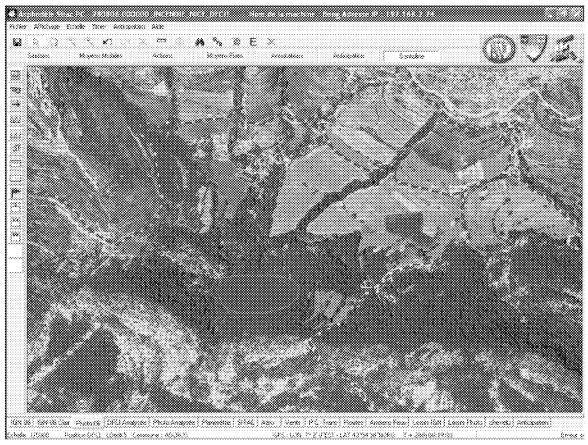


Figure 6: Initial fire area © SDIS06

2. The current dominant wind is specified and therefore the corresponding matrix is fixed. Then, the shade temperature, the soil water content and the global wind velocity are typed in. By means of the Thornthwaite formula, a rate of spread is calculated. At this point, the result can be interpreted and, if necessary, it is corrected. Indeed, the firemen involved in the project have great experience in the use of the Thornthwaite formula. For instance, they know that, right after a rainfall, the rate of spread is slightly over-evaluated while it is under-evaluated in case of severe drought. Moreover, corrections of the global estimated speed can also be made on the basis of observations made on the field.

3. The areas that may be reached by the fire during the simulation are classified. Thus, each of them is assigned one of the three following values: non-flammable, low-flammable or flammable (cp. Figure 7). Then, this information is passed on the array LD: this requires an interaction between the global procedure and the underlying GIS that manages the map's display.

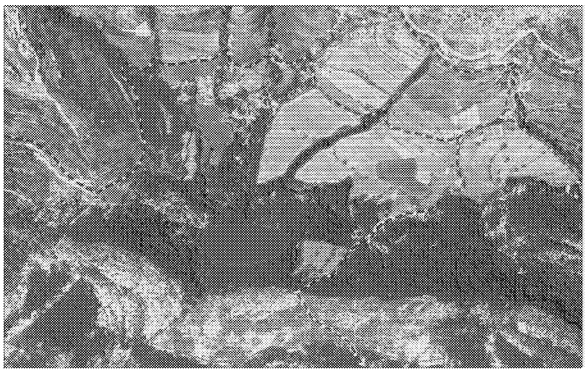


Figure 7: Classification of certain areas © SDIS06

4. For each cell of LD being on the outline of the fire area, the *local* wind direction and velocity is provided by the wind matrix which may therefore be regarded as an abacus. If necessary, the actual local wind velocity W_v is fixed by means of a rule of three. Then, the Thornthwaite formula can be applied: this enables us to determine the *local* rate of spread main vector hereafter denoted R (cp. Figure 8).

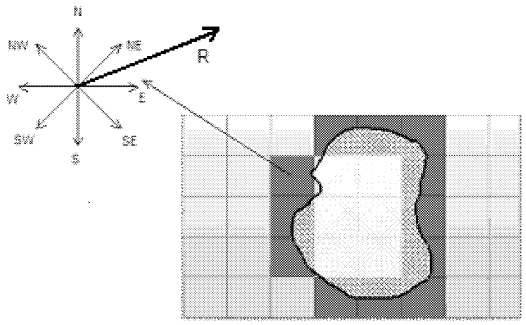


Figure 8: Propagation phase

At this point, the vector R is projected on each of the four axes [N-S], [E-W], [NW-SE] and [NE-SW]. These are but the vertices that link the current cell to its height neighbouring cells (cp. Figure 8). Thus, we have now four different rates of spread, denoted R_1, R_2, R_3 and R_4, each of them corresponding to a given vertex. Then, these rates are respectively modified in order to take into account the slope coefficient proper to each vertex. Moreover, the effect of the level of flammability is expressed by means of specific coefficients too. On this subject, let us note that the first half of the trajectory is inside the source cell whereas the second half is inside the target cell. In both cases, the right coefficient must be applied.

On the basis of the above data, the duration required for fire propagation from a source cell to a target cell along a given vertex can be calculated. Of course, a given target cell is normally reached from several source cells. Hence, among the different durations, only the shortest one is relevant. Moreover, if a target cell belongs to a non-flammable area, the duration is replaced by a specific value which indicates the fact that the cell will never be reached. When the above process has been performed for *all* the cells belonging to the initial fire area outline, it is repeated for all the target cells, and so on … This process generates successive generations of cells and provides the minimum length of time the fire needs to reach each of them.

5. The final area of the fire corresponding to a given duration can be defined by the set of cells reached within this delay (cp. Figure 9). Insofar as we do not want to under-predict the fire's spread, we may also choose to define the burning area by the set of cells adjacent to the previous ones.

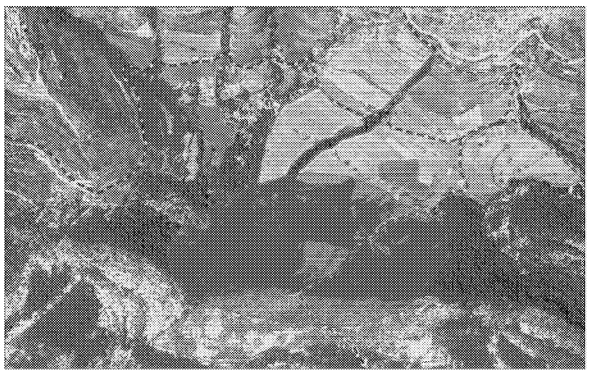


Figure 9: Fire spreading after 1h © SDIS06

PERFORMANCE

The experimental phase is carried out using an Intel-based Centrino Core Duo 1,8 Ghz portable computer with 2 Go RAM. Four different sizes (from 500 ha up to 50000 ha) have been defined as initial fire surfaces. Then, the final fire's shape was calculated on the basis of three different durations, namely 30 mn, 2 hours and finally 4 hours:

Surface \ Duration	30 mn	2 h	4 h
500 ha	**30 s**	**45 s**	**01:10 s**
1000 ha	**50 s**	**01:05 s**	**01:18 s**
5000 ha	**01:30 s**	**01:50 s**	**02:30 s**
50000 ha	**12:00 s**	**13:20 s**	**14:08 s**

The experimental phase leads to the following comments:

1. First, we note that most of the time of computation is dedicated to area classification on the map and to the transfer of this information to the array LD.

2. Without the area classication phase, the length of time of the different simulation phases would not exceed five minutes: such a delay totally fulfils our requirements in terms of timing constraints. Moreover, wide areas of landscape can be processed without any problem.

CONCLUSION AND FUTURE WORKS

The software system described in this paper is in test phase since the beginning of the summer 2007 and the first results are quite promising. So far, we have noticed the following points:

1. The level of interoperability with the other Asphodèle's software components is satisfying.

Moreover, the experience of the firemen can be taken into account by the system.

2. Because of the lanscape discretization by means of regular cells, distortion of the fire shape can occur, and it should be advisable to cope with this problem.

3. The performances are in conformity with the timing constraints imposed by the framework of fire suppression command-line.

Moreover, a huge amount of work is still required to turn the system into a fully operational tool. Therefore, future works will first and foremost focus on:

1. The validation of the respective values of the parameters that appear in the equations defining the influence of the slope. It is worth noticing that these coefficients have generally been defined for climatic areas that are totally different from the mediterranean one.

2. A more precise classification of the different areas of the landscape: as a matter of fact, we already have maps that specifies the different kinds of vegetation. Consequently, we intend to distinguish five different levels of flammable areas according to the different vegetable species and to the density of the vegetation. We also intend to take into account the effect of sun exposure on vegetation moisture content.

3. The areas classification phase must be subject to a preprocessing phase using the aforesaid maps: this will drastically reduce the duration of the simulation phases.

REFERENCES

Bodrozic, L., Stipanicek, D., and M. Seric. "Forest fires spread modelling using cellular automata approach". Department for Modelling and Intelligent Systems. University of Split, Croatia, 2006

Cheney, N.P. "Fire Behaviour". In: *Fire and the Australian biota.* Australian Academy of Science. pages 151-175, 1981

Dimitrakopoulos A.P. and S. Drita. "Novel nomographs for fire behaviour prediction in Mediterranean and submediterranean vegetation types". In: *Forestry,* Vol.76, No.5, pages 479-490, 2003

Dumond, Y. "Modelling forest fires fight by means of a numerical battlefield". In Proc. 20th European Simulation and Modelling Conference. 23-25 October. Toulouse, France, 2006

Finney, M.A. "FARSITE : Fire Area Simulator-model development and evaluation" . US Department of Agriculture, Forest Service, Rocky Mountain Research Station. Paper INT-4. Ogden, UT, USA, 1998

Forestry Canada Fire danger Group. "Development and structure of the Canadian Forest Fire Behaviour Prediction System". Information Report ST-X-3. Ottawa, Canada, 1992

Martins Fernandes, P.A. "Fire spread prediction in shrub fuels in Portugal". In: *Forest Ecology and Management,* No.144, pages 67-74, 2001

Nelson, M.N. Jr. "An effective wind speed for models of fire spread". In: *International Journal of Wildland Fire,* Vol.11, pages 153-161, 2002

Perry, G.L.W. "Current approaches to modelling the spread of wildland fire: a review". In: *Progress in Physical Geography*, No.22, pages 222-245, 1998

Rabner, K.W., Dwyer, J.P. and B.E. Cutter. "Fuel model selection for BEHAVE in midwestern oak savannas". In: *Northern Journal of Applied Forestry,* Vol.18, No.3, pages 74-80, Jamestown, Northern Prairie Wildlife Research Center Online, 2001

Richards, G.D. "The properties of elliptical wildfire growth for time dependent fuel and meteorological conditions". In: *Combustion, Science and Technology*, No.95, pages 357-383, 1994

Rothermel, R.C. "A mathematical model for predicting fire spread in wildland fuel". US Department of Agriculture Forest Service, Intermountain Forest and Range Experiment Station General Technical Report INT-115, Ogden, UT, USA, 1972

Thornthwaite, C.W. "An approach toward a classification of climate". In: *The Geographical Review,* No.38, pages 55-94, 1948

Van Wagner, C.E. "Effect of slope on fires spreading downhill". In: *Canadian Journal of Forest Research,* Vol.18, No.6, pages 818-820, 1988

Viegas, D.X. "Slope and wind effects on fire propagation". In: *International Journal of Wildland Fire,* Vol. 13, pages 143-156, 2004

Weise, D.R. and G.S. Biging. "A qualitative comparison of fire spread models incorporating wind and slope effects". In: *Forest Science,* Vol. 43, No.2, pages 170-180, 1997

ACKNOWLEDGEMENT

The author wishes to thank Mrs Eryl-Anne Baylis for their careful review of this paper and whose comments have been a great help in improving the English style of the text.

BIOGRAPHY

YVES DUMOND obtained his PHD in Computer Science from the University of Nice Sophia-Antipolis in 1988. He is currently associate professor at the University of Savoie (France), where he has worked since 1990. In particular, he has been involved in the Eurêka project PVS 2434. His main topics of interest include formal specifications and concurrency theory, especially process calculi.

FLUID FLOW SIMULATION

CANDU 6 NUCLEAR POWER PLANT CONTAINMENT MODELING

Ilie Prisecaru and Daniel Dupleac
Power Plant Engineering Faculty
Politehnica University of Bucharest
313 Splaiul Independentei, 060042, sector 6, Bucharest, Romania

E-mail: prisec@cne.pub.ro, danieldu@cne.pub.ro

Mihaela Biro
Power Plant Engineering Faculty
Politehnica University of Bucharest
313 Splaiul Independentei, 060042,
sector 6, Bucharest, Romania

KEYWORDS
Nuclear engineering, Model development, Model evaluation, Lumped parameter, Continuous simulation

ABSTRACT

The paper presents the mathematical model and thermal-hydraulic analysis of CANDU Nuclear Power Plant (NPP) containment. The mathematical model for the containment behavior was developed. This mathematical model was coded in ACSL simulation language. To validate the mathematical model, 80 % break in Reactor Outlet Header (ROH) Loss of Cooling Accident (LOCA) was simulated and the results were compared with those of PRESCON code. The paper presents the mathematical model, the assumption made and the analyses of simulation results.

INTRODUCTION

One of the basic features of the nuclear safety is the provision of a number of inherent and special barriers to the release of radioactive materials from the fuel to the environment. Over 99.9% of the radioactivity produced by the operation of a nuclear power plant is held in the fuel. The fuel itself is an inherent barrier, the fission products for the most part are trapped within the fuel material. The fuel sheath and the intact primary coolant system serve as the second and third barriers to the release of radioactive materials to the containment. The failure of these components could result in the release of significant amounts of radioactivity within the containment envelope, which is the third barrier. The final barrier is the exclusion zone which surrounds the reactor. The public is not permitted to be within this area. This zone provides dilution of the dispersed activity.

Containment is the structural envelope around the nuclear components of the plant and one of the special safety systems that prevents or limits activity releases to the environment. The containment envelope is the principal component of the containment system. It surrounds the space where highly radioactive atmosphere may be present, and includes the penetrations of the walls. It is made of prestressed reinforced-concrete to meet the design goals. The containment walls and roof provide radiation shielding for the outside environment, and, obviously, normal housing and weather protection for the reactor. There are many components of the containment system in order to complete the desired functions. Between these, the components that are directly related to pressure reduction inside the containment are the dousing system for pressure suppression, steam condensation, atmosphere cooling and absorbing, i.e. washing out, some airborne radioactive materials and the local air coolers for containment atmosphere cooling (AECB, 1993).

The design pressure of the containment envelope is the upper bound of the peak pressures resulting from all the design basis accidents considered. For the CANDU plant the design containment pressure is based on primary coolant circuit LOCA and is relatively low compared to the design pressures for the containment envelope for other reactor types. To meet this pressure criteria the local air coolers, condensation on the containment walls and the dousing spray system all help to reduce the severity of the pressurizing accident by absorbing energy.

Thermal hydraulic analysis of containment must certify that the containment design pressure is higher than the calculated peak value of overpressure in design basis accidents. However an overpressure over the containment design pressure will not lead to the containment complete failure. Instead, cracks will develop in the concrete walls of the containment and the leakage rate from the pressurized containment atmosphere through the perimeter walls of the reinforced concrete containment structure will increase. This will intensify the release of radioactive gases and vapors following and accident.

The paper underlines the containment model development and results obtained for the pressure evolutions after a Loss of Cooling Accident (LOCA) are presented.

MODEL DESCRIPTION

The model developed is achieved through the application of first principle and can simulate containment response under both normal and abnormal condition. The mass and energy balances equations are performed for each control volume (or node) and conservation of momentum is used to determine the flow rates between control volumes.

The number of control volumes and their location depend on the physical structures and restrictions, equipment and their location, and the existing flow paths. Thus the number of control volumes is plant dependent. The number of nodes was derived as minimum required bettering representing the pressure distribution inside the containment taken into account the above mentioned restrictions.

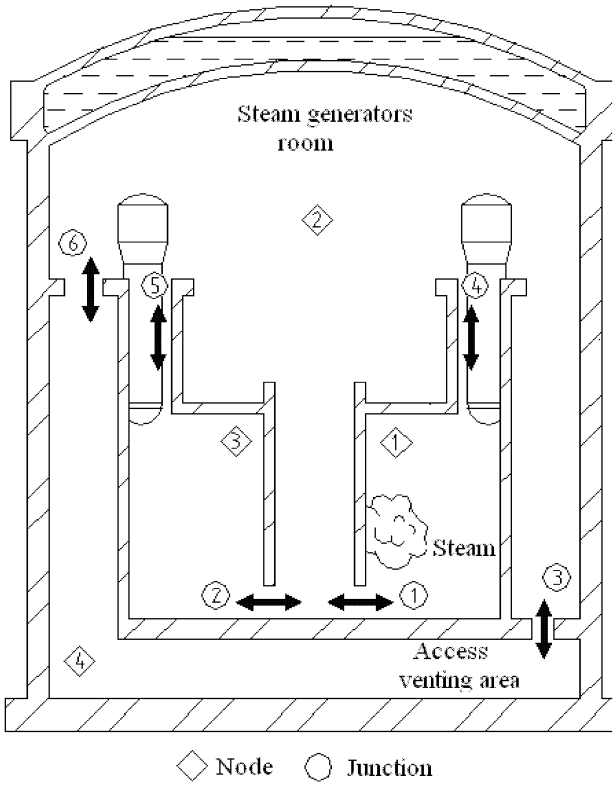


Figure 1: CANDU-6 Containment Control Volumes

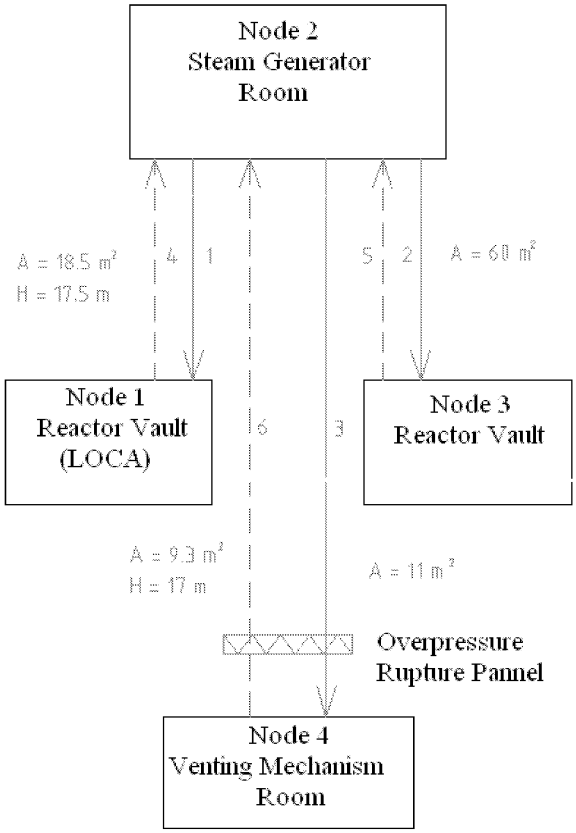


Figure 2: Considered Containment Flow Path

As result, 4 nodes and 6 junctions was considered for the base model, as shown in Fig. 1 and Fig. 2. The nodes considered are: the Steam Generators Room (upper volume of the containment), Reactor Vault – two nodes, one node at each side of reactor – and Access Venting Area zone.

The main assumptions made at the model development are the following:

- Lumped parameters are considered for each node;
- Noncondensibles gases are not modeled;
- Air is considered an ideal gas with constant heat capacity at constant pressure;
- Superheated steam is also considered as an ideal gas;
- Subcooled water is incompressible and its thermodynamic properties are calculated at saturation point;
- Steam-air mixture is considered homogeny and in thermal equilibrium;
- Heat losses through equipments, pipes, etc. located inside containment are considered as heat sources. They can be treated as constant or time-dependent variable;
- Leaks through containment wall are neither modeled;
- The heat extracted by the local air coolers is dependent on the humidity of the steam-air mixture;
- Heat losses through wall condensing are taken in account. The heat transfer coefficient is dependent on the humidity of the steam-air mixture. In the case in which the containment walls became heat sources, the heat transfer coefficient is calculated as in natural convection case;
- The energy balance equations take into account the heat loosed through the water of the dousing system. However, this water is not taken into account in the mass balance equations;
- As the pressure in the reactor pipe systems are much higher than the pressure inside of containment and the fluids inside pipe are hot, all the water entering the containment through a break suddenly will vaporize.

Conservation of Mass

A mass balance is performed for each node considering each possible flow path. For the total mass (steam and air) this equation is:

$$\frac{dM_{Ti}}{dt} = \pm \sum_{k \in K} W_k + W_{Ti} \qquad (1)$$

The equation (1) is considered together with the air mass balance equation for each control volume:

$$\frac{dM_{Ai}}{dt} = \pm\sum_{k\in K}\left(\frac{M_A}{M_T}\right)_{Li} W_k + W_{Ai} \quad (2)$$

Conservation of Energy

The internal energy balance is applied at each control volume and takes into account all the relevant inflows and theirs enthalpies, as well as heat fluxes and heat exchange with heat sources and heat sinks:

$$\frac{dU_{Ti}}{dt} = \pm\sum_{k\in K} h_b W_k + Q_i \quad (3)$$

Conservation of Momentum

The mass flow rate between internal control volumes is obtained from the conservation of momentum equation. Two forms of momentum equations are used depending on junction geometry. The first form used considers the case of upward flow due to the variation in density of control volumes and is similar to flow through orifices. Thus, in this particular case (flow path marked with dotted line in Fig. 2), the flow rate is given by the pressure differences between adjacent nodes:

$$\Delta p_{ij} = p_i - p_j + (\rho_i - \rho_j) gH \quad (4)$$

and

$$W_{ij} = C_{ij}\sqrt{\rho_i \cdot \Delta p_{ij}} \quad (5)$$

The second form considered is used when the flow path can be considered as a horizontal channel. The conservation of momentum equation is:

$$\frac{1}{A}\frac{\delta W}{\delta t} = -\frac{1}{A}\frac{\delta}{\delta x}\left(\frac{W^2}{\rho A}\right) - \frac{\delta p}{\delta x} - \frac{1}{A}\frac{\delta F}{\delta x} \quad (6)$$

Integrating the equation (6) between the center of two adjacent nodes, i and j, and after some algebra, we finally find:

$$\frac{dW_{ij}}{dt} = \frac{1}{I_{ij}}\left[W_{ij}^2\left(\frac{1}{\rho_i A_i^2} - \frac{1}{\rho_j A_j^2}\right) + \frac{W_{ij}^2}{\rho_i + \rho_j}\left(\frac{1}{A_j^2} - \frac{1}{A_i^2}\right) + (p_i - p_j) - K\frac{W_{ij}^2}{(\rho_i + \rho_j)A_j^2} - \frac{f_i L_i}{4d_i}\frac{W_{ij}^2}{\rho_i A_i^2} - \frac{f_j L_j}{4d_j}\frac{W_{ij}^2}{\rho_j A_j^2}\right] \quad (7)$$

where

$$\int_j \frac{dx}{A} = \int_{j1}^{i1} \frac{dx}{A} + \int_{j1}^{j} \frac{dx}{A} = \frac{L_i}{2A_i} + \frac{L_j}{2A_j} = I_{ij} \quad (8)$$

Temperature and Pressure Calculation

Assuming perfect mixing in each control volume, the state of each control volume can by found having previously determined the mass and internal energy of each node.

The internal energy of air-steam/water mixture is simply the sum of the internal energy of the individual component (Wark, 1983):

$$U_T = (M_T - M_A)\cdot u + M_A \cdot c_{vA} \cdot (T + 273.15) \quad (9)$$

The temperature is found using equation of state through a procedure that can be found in (Toderas, 1993). Having determined the temperature, the partial pressure of individual components in each control volume is computed. The air partial pressure is calculated from the perfect gas law (Wark, 1983):

$$p_A = \frac{M_A \cdot R_A \cdot (T + 273.15)}{V} \quad (10)$$

whereas, the partial pressure of the vapor is calculated as a function of density and temperature. Then, the total pressure of each control volume is calculated as the sum of partial pressure of individual components:

$$p = p_A + p_v \quad (11)$$

Heat Sources and Heat Sinks

To close the energy conservation equation (3), the heat sources and sinks term must be provided. The main heat source is the heat entering from the break of reactor system piping. The amount of heat contributing to pressurize the containment is simply calculated from the enthalpy and the flow rate of fluid discharged through break and is a user input in the model and is dependent of the accident analyzed. Also as heat source we have the contribution of heat losses through equipments, pipes, etc. located inside containment. These are also a user input in the model.

The mains heat sinks considered are:

- Heat extracted by local air coolers: this term is calculated as function of air humidity and cooling water temperature;
- Heat extracted by dousing water: this term is calculated as function of dousing flow rate and droplet temperature increase;
- Heat transferred to the containment structure walls by condensation (Collier, 1994). In order to find the wall temperature, one-dimensional time-dependent conduction equation (Holman, 1981) is solved assuming a constant temperature for the outside containment air temperature.

RESULTS

The mathematical model was coded in ACSL simulation languages (MGA, 1993). To validate the mathematical model and his implementation in ACSL a number of transients representing pipes breaks inside containment were investigated. The results were compared with those obtained with the PRESCON code (Elliott, 1971). PRESCON was used for containment behavior in safety analysis of CANDU reactors (Cernavoda, 1996).

The LOCA following transient presented in this paper represent the containment response after an 80% break of Reactor Outlet Header. The break flow and enthalpies was input data and was taken from (Cernavoda, 1996).

The Fig. 3 shows the containment response during the LOCA. The figure shows the containment pressure versus time after rupture predicted by the above mathematical model and PRESCON code.

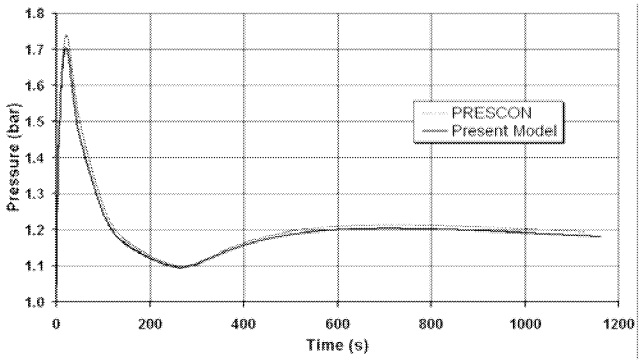


Figure 3: Containment pressure transient

The pressure rapidly increases due to the large amount of heat released inside containment. The peak pressure predicted is 1.705 bars attained at 21 s after break occurs compared with 1.74 bars attained also at 21 s predicted by PRESCON code. At 1.15 s the dousing initiation signal is reached but dousing became effective after 6.15 s. The value of the peak pressure depends of the amount of energy provided from the reactor coolant system and the amount of energy that is transferred through the water droplets, through the local air coolers heat exchangers, and through the wall by condensation. The containment pressure decrease significantly when the energy extracted from containment atmosphere surpasses the heat added by water discharged through the break.

Since the containment pressure remain above 1.07 bars, the dousing system is continuous operating, thus after 301s the dousing system is emptied of water. As a result, a second rise in containment pressure results. However, this increase in pressure is slower than immediately after the break due to smaller amount of heat released inside containment through break during this period of accident.

Figure 4 shows the temperature transient inside containment. The maximum temperature is attained in the Reactor Vault where the header is broken. This maximum temperature value is 115.2 ^{o}C reached at 21 s after the break. The Steam Generators Room peak temperature is 84.7 (at 20 s) whereas in Access Venting Area the maximum temperature is 48.9 ^{o}C (at 25 s).

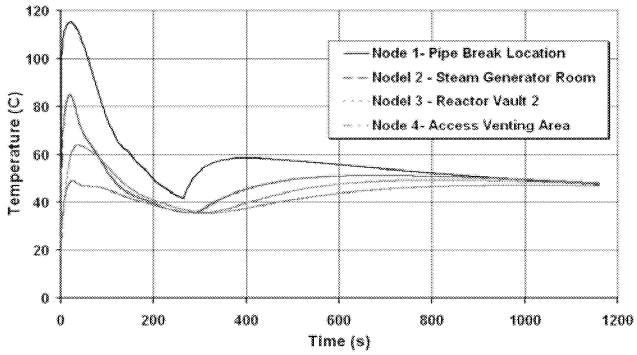


Figure 4: Containment temperature distribution

At the end of the transient, the control volumes temperature is practically the same, 47 ^{o}C. Similar results were obtained also with the PRESCON code. The results show the successful implementation of the mathematical model to correctly determine the pressure and temperature response inside the containment following a LOCA.

The same transient was run in the case of the dousing system unavailable. Fig. 5 shows the containment pressure whereas the Fig. 6 shows the temperature variation for this particular case.

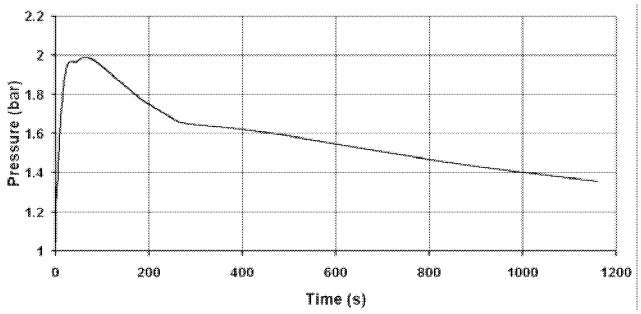


Figure 5: Containment pressure transient when dousing system is unavailable

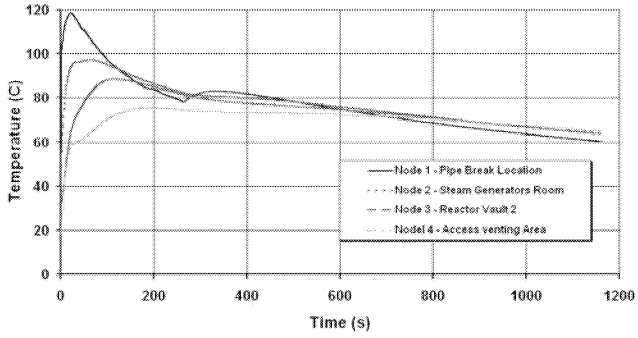


Figure 6: Containment temperature distribution transient when dousing system is unavailable

When dousing system is unavailable, the containment peak pressure is 1.988 bars and is reached at 67 s after the transient. Since unavailability of an important heat sink, the maximum pressure reached is higher and the pressure decreases is slower in this case, as expected. This shows the importance of dousing in reducing the maximum attainable containment pressure as the heat transferred through water droplets is a very efficient mechanism of heat removal. Although this mechanism for energy removal acts only for the first 301s, it is the main heat sink for the period of accident analyzed, as shown in Fig. 7.

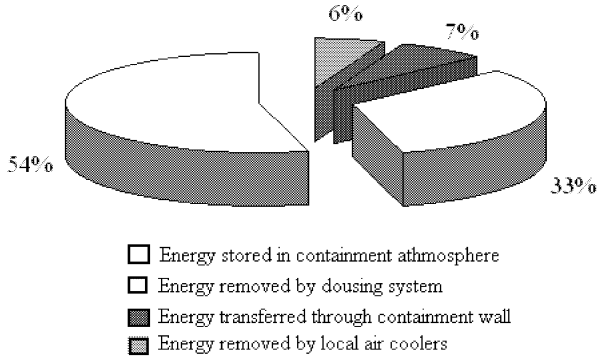


Figure 6: Energy removal contribution of each heat sink

The temperature in the Reactor Vault where the header is broken reaches 118.5 ^{o}C after 22 s subsequent to break. Much higher temperature is also attained in other control volumes: 97.1 ^{o}C for Steam Generators Room (at 66 s), 84.5 ^{o}C for the other Reactor Vault (at 118 s) and 75.4 ^{o}C for Access Venting Area (at 201 s)

CONCLUSIONS

In this paper a mathematical model for CANDU 6 containment is presented. The main aspects of the mathematical model are highlighted. The mathematical model was coded in ACSL simulation language. The ACSL coded program was used for the validation of the mathematical models. The transient shown in the paper represent an 80% break in Reactor outlet Header. The results obtained are compared with those obtained with the PRESCON code. The comparison shows that the mathematical model can correctly predict the temperature and pressure inside containment and was properly coded.

As further work, the models will be implemented in the Modular Modeling System (MMS) code using the CompGen tools (Framatome Tehnologies, 1998), improving the library of CANDU models for the MMS package (Prisecaru, 2004).

REFERENCES

Atomic Energy Control Board (AECB), 1993. *Fundamentals of Power Reactors – Module Two: Nuclear Systems*. Training Manual.

Cernavoda Nuclear Power Plant, 1996. *Final Safety Analysis Report*. Cernavoda NPP Unit 1, FSAR Chapter 15.

Collier, J.G.; and J.R.Thome. 1994. *Convective Boiling and Condensation*, Clarendon Press.

Elliott, J.N.; Odlozinski, G; and D'Arcy, D.F. 1971. *PRESCON: A Digital Computer Program for Containment Pressure Transients*, Proceedings of conference on new developments in reactor mathematics and applications, Idaho Falls, USA.

Framatome Tehnologies. 1998. *MMS Theory Manual Release 5.1*.

Holman, J.P. 1981, *Heat transfer*, Mc-Graw Hill, New York.

Mitchell and Gauthier Associates (MGA) Inc. 1993. *Advanced Continuous Simulation Language (ACSL) Reference Manual*. Edition 10.1

Prisecaru, I.; D. Dupleac; and Constantinescu, A. 2004 "CANDU Steam Generators Modeling Using CompGen for MMS Package". *Proceedings of international conference ESMc' 2004 European Simulation and Modelling Conference)*, 25-27 oct, Paris, France

Todreas, N.E.; and M.S. Kazimi 1993, *Nuclear Systems, Thermal hydraulic Fundamentals*, Taylor &Francis.

Wark, K. 1983, *Thermodynamics*, Mc-Graw Hill, New York.

NOTATION

A – flow area, m^2
C – hydraulic conductance, m^2
c_v – heat capacity at constant volume, J/kg K
d – hydraulic diameter, m
F – force, N
f – friction coefficient
g – garvity acceleration, m/s^2
H – height, m
h – enthalpie, J/kg
K – local energy losses coefficient
L – lenght, m
M – mass, kg
p – pressure, Pa
Q – thermal power, W
R – gas constant, J/kg K
S – surface, m^2
T – temperature, ^{o}C
U – internal energy, J
u – specific internal energy, J/kg
V – volume, m^3
W – flow rate, kg/s
Δp – pressure loss, Pa
ρ – density, kg/m^3

Subscripts

A – air
b – break
i – node numbering index
j – node numbering index
k – junction numbering index
T – total (steam+air)

An evaluation of the influences of compressibility, body forces, rheology and wall slip on the computer modelling of injection moulding of thermoplastics

Vahid Nassehi
Abhijeet Kulkarni
Department of Chemical Engineering
Loughborough University
Loughborough LE11 3TU
Email: v.nassehi@lboro.ac.uk

Ramin Salemi
EIS Climate Division
Visteon Corporation
Visteon Customer &Technology Centre
Essex SS14 3WF
E-mail Address: rsalemi@visteon.com

KEYWORDS

Injection Moulding, Thermoplastics, Body forces, Wall slip, Computer modelling, rheology

ABSTRACT

Computer modelling of injection moulding of polymeric material is a long established technique for the quantitative analysis of the performance of different types of moulds. This is because that it provides a powerful and convenient methodology for the prediction of the outcome of injection moulding processes. Therefore this technique has been frequently used by the design engineers to develop new types of moulds as well as the moulding processes. The injection moulding of polymers is normally carried out under large pressures and hence the effects of other forces are normally assumed to be negligible. This may be an acceptable assumption in many situations, however, there is some experimental evidence that show the inadequacy of such approach. In this paper we examine the effects of various types of body forces on the modelling of injection moulding of thermoplastics. Other effects such as wall slip, compressibility and elastic behaviour of polymers are also investigated. In order to show that the obtained results are not dependent on the choice of any particular numerical scheme we have used both a well known commercial simulation package based on the finite volume method as well as an in-house developed finite element based software.
Our results prove that the accuracy of the computer simulation of injection moulding improves significantly if relatively minor effects such as the influence of body forces and slight compressibility of polymers are included in the governing equations of the mathematical model of the process.

INTRODUCTION

As already mentioned computer modelling has been extensively used in the simulation of injection moulding of thermoplastics (e.g. see Isayev, 1991). However, despite the wide spread application of this technique quantitative evaluation of the influence of relatively minor factors on the modelling accuracy has received little attention. This is because that injection moulding of polymers, which is carried out under large pressures, is thought to be entirely governed by the rheological behaviour of such material in a pressure induced flow. Although, rheology plays a very significant role in the outcome of injection moulding of polymeric material (Hanspal, *et al*, 2005) it is not the sole factor affecting the process. In particular, in dealing with thermoplastics in which elongation and other memory effects are not always dominant features of a flow process, factors such as degree of compressibility of the polymer and body forces may need to be taken into account. Representation of such phenomena in mathematical flow models is relatively straightforward. However, the solution scheme that is used to solve such models should be selected carefully. For example, as is shown later in this paper, unless slippage of the polymeric fluid on the mould walls is taken into account an injection moulding model will not yield accurate simulation results. Another important consideration is the selection of moving boundary tracking method in an injection moulding scheme. A wide variety of techniques are available which can be used to simulate mould filling process (Nassehi, 2002). However, to preserve the accuracy of simulations in choosing a technique to develop a moving boundary tracking scheme for injection moulding process factors such as speed of the operation, geometry of mould cavity, rheological properties of the polymeric fluid and the ratio of gravitational to viscous forces (Nassehi and Petera, 1994) should be considered. In the following sections of this paper after the description of the mathematical model used in this study the obtained computational results are presented and discussed. These discussions reveal the influence of the so called minor effects on the outcome of the simulation of injection moulding of thermoplastics.

MATHEMATICAL MODEL

The continuity equation representing conservation of matter used in the present model is based on the following mass balance equation for incompressible fluids

$$\nabla \cdot \mathbf{V} = 0 \qquad (1)$$

Where $\mathbf{V}$ is the velocity vector. This equation is modified to the following perturbed form to include the slight

compressibility of thermoplastic material in the mathematical model.

$$\frac{1}{\rho c^2}\dot{p} + \nabla \cdot \mathbf{V} = 0 \qquad (2)$$

Where ρ is the fluid density, c is the speed of sound in the fluid and $\dot{p}$ is the temporal derivative of the pressure. Therefore in equation (2) the effect of compressibility of a slightly compressible fluid is represented in terms of a relationship between its dilatation, bulk modulus and the pressure under which the flow is taking place. Further to the freedom gained to investigate the influence compressibility on the accuracy of the simulations it should also be noted that the addition of this term to the continuity equation improves the stability of the numerical schemes used to solve it (Zienkiewicz and Wu, 1991).

In conjunction with an appropriate mass balance relationship the flow dynamics in an injection moulding process is represented by the following momentum balance equation.

$$\rho\frac{\partial \mathbf{V}}{\partial t} + \rho\mathbf{V}\nabla \cdot \mathbf{V} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho\,\mathbf{f} \qquad (3)$$

Where $\mathbf{V}$ is the flow velocity, p is pressure, $\boldsymbol{\tau}$ is the extra stress tensor and $\mathbf{f}$ is the body force vector per unit volume of fluid. Equation (3) yields the Navier-Stokes equation for laminar free flow of a generalised Newtonian fluid if the stress term in the right hand side can be replaced in terms of viscosity and velocity gradients (i.e. rate of deformation) tensor. For a purely Newtonian fluid (i.e. a linear Stokesian fluid) where viscosity is a constant μ, the stress term can simply be written as $-\mu\nabla^2\mathbf{V}$. For a generalised Newtonian fluid this term is represented in terms of η (stress dependent viscosity) as $\nabla\eta\nabla.\mathbf{V}$. In this study we have considered two options. The first option is the representation of thermoplastic polymer as a non-elastic shear thinning fluid. In this case the stress dependent viscosity is defined in terms of the following power-law model

$$\eta = \eta^o\left(\dot{\gamma}\right)^{n-1} \qquad (4)$$

Where, η^o is the material consistency coefficient, n is the power-law index ($0 < n < 1$ for a shear thinning fluid) and $\dot{\gamma}$ is the shear rate (De Waele and Ostwald, 1925). The second option considered in this study is to include elastic effects in the model by the use of Criminale-Erickson-Fibley (CEF) constitutive equation (Bird *et al*, 1987). In this model the stress is related to the strain rate via the following equation.

$$\tau = \eta\left(\dot{\gamma}\right)\dot{\gamma} + \left(\frac{1}{2}\psi_1 + \psi_2\right)\left\{\dot{\gamma}\cdot\dot{\gamma}\right\} - \frac{1}{2}\psi_1\frac{D\dot{\gamma}}{Dt} \qquad (5)$$

Where ψ_1 and ψ_2 are material parameters known as the first and second normal stress difference coefficients, which are also functions of the magnitude of shear rate, i.e. $\left|\dot{\gamma}\right|$.

The temporal derivative appearing in the right hand side of equation (5) is defined as,

$$\frac{D\dot{\gamma}}{Dt} = \frac{d\dot{\gamma}}{dt} + \boldsymbol{V}\cdot\nabla\dot{\gamma} + (\nabla\boldsymbol{V})^T\cdot\dot{\gamma} - \dot{\gamma}(\nabla\boldsymbol{V}) \qquad (6)$$

In cases where the material parameters ψ_1, ψ_2 and η are constants, the CEF model reduces to a simpler expression and for a shear dominated flow process such as injection moulding these parameters can be determined experimentally. In this study we have used the following relationships for the normal stress difference coefficients (Nassehi and Salemi, 1994).

$$\begin{cases} \psi_{12} = A\eta^b\left(\frac{1}{2}I_2\right)^{\left(\frac{b-2}{2}\right)} \\ \psi_{23} = \frac{1}{10}\psi_{12} \end{cases} \qquad (7)$$

Where A and b are two characteristic constants of a polymeric fluid and I_2 is the second invariant of the rate deformation tensor (i.e. the shear rate $\dot{\gamma}$)

Using both options the stress term in the equation of motion can be replaced in terms of velocity gradients and material parameters. Hence the system of governing equations becomes determinate and can be integrated (solved) subject to appropriate boundary conditions.

BOUNDARY CONDITIONS

Apart from the inlet section, where the flow rate or inlet velocity can be imposed as an essential (i.e. Direclet) boundary condition the prescription of boundary conditions along the other sections of the domain representing an injection moulding process is not straightforward. In this respect, the imposition of no-slip conditions on the solid walls, although simplifying the modelling procedure, is found to be incompatible with the treatment of the advancing flow front as a free surface boundary. Therefore, to maintain the accuracy of the simulations partial slip on the solid walls of the mould need to be imposed. This is achieved through the imposition of the following Navier's wall slip condition (Silliman and Scriven, 1980).

$$\left[\beta\boldsymbol{\tau}.\mathbf{n} + (\mathbf{V} - \mathbf{V}_b)\right]\cdot\mathbf{t}^T = 0 \qquad (8)$$

Where $\mathbf{n}$ and $\mathbf{t}$ are, respectively, unit vectors normal and tangent to the solid boundary, β is a friction (slip) coefficient (dependent on the physical properties of the

solid walls) and $\mathbf{V}_b$ is the velocity of the solid walls (in the case of injection moulding process $\mathbf{V}_b$=0). Equation (8) together with the wall condition of

$$\mathbf{V} \cdot \mathbf{n} = 0 \tag{9}$$

provide the realistic wall boundary conditions for a mould filling process. To incorporate the slip wall condition a narrow slip layer in which equations (8- 9) instead of the normal equation of motion are solved is used (Ghoreishy and Nassehi, 1997).

The moving free surface front is simulated using the VOF (volume of fluid) technique. This method is based on the solution of the front position's probability density equation, given as

$$\frac{\partial F}{\partial t} + \mathbf{V} \cdot \nabla F = 0 \tag{10}$$

Where $0 \le F \le 1$ is defined as the free surface position function. As mentioned earlier, in this study we have used both commercial and in house developed software in our simulations. In the finite element technique used to develop our own solution scheme we have used a modified version of the VOF technique in which the moving front flow is considered as a two-phase regime in which the wet and the dry sections represent the phases (Thompson, 1986). The flow model is solved for the entire domain whilst at each section physical properties relevant to that phase are inserted. A set of fictitious physical values that render the velocity in the dry regions as zero are used and hence the technique is called the pseudo-density method. Simultaneous solution of equation (10) with the flow model generates values of F. A value of F between 0 and 1 (usually $F = 0.5$) is taken as the boundary between phases representing the moving front. Values of physical parameters for each phase in the flow field is related to the position of the front using the following equation

$$y = y_w F + y_d \left(1 - F\right) \tag{11}.$$

Where y is a given physical parameter and y_f and y_a are the values of this parameter in the wet and dry regions, respectively, (Nassehi and Ghoreishy, 1997).

SOLUTION SCHEMES

The commercial software used in this study are the well known *FLUENT* and *STAR CD*. These packages use the finite volume method and can generate three dimensional simulations whilst maintaining computing economy. There is a large body of published literature which describe the theoretical and practical aspects of the implementation of the finite volume method (e.g. see Patankar, 1980; Versteeg and Malalasekera, 1995) and hence these details are not discussed here. However, the use of a commercial package for a research study is inherently restrictive as it does not allow any significant alteration of the basic solution schemes or the governing model. To resolve this problem parallel with the use of commercial packages we have developed a finite element based scheme to solve the described governing equations of the injection moulding process. This scheme has allowed us to readily investigate the influences of factors such as wall slip and inclusion of elastic effects via the CEF model on the outcome of the mould filling simulations of thermoplastics. The finite element scheme used depends on the discretization of the mathematical equations via a mixed UVP technique (Two dimensional Cartesian version where U,V & P represent velocity components and pressure respectively). As these details are published elsewhere (e.g. see Nassehi, 2002) they are not repeated here.

SAMPLE RESULTS AND DISCUSSIONS

According to the available experimental evidence the geometry of advancing free surface front inside the mould during an injection moulding process is noticeably affected by relatively minor alterations in the physical properties of the thermoplastic being moulded and the conditions of the mould itself (Hanspal, *et al,* 2005). Therefore the main criteria used for the comparison of the simulation results obtained in this work and the evaluation of their accuracy and compatibility with the realistic situations has been the shape of the advancing front. We start with the study of results generated using the *STAR CD* package. These results are generated using the set of data given in Table 1.

Boundary Conditions:
Inlet velocity = 0.1 ms^{-1} normal to the inlet region of 7 mm width;Outlet pressure = 0 Pa (gauge) at 50 mm width outlet region; Solid walls – no slip, stationary walls.
***Initial Conditions*:**
Free surface was set at the bottom of the 50 mm wide rectangular area;
Velocity components and pressure were initialised to zero.
***Fluid Properties*:**
Density, ρ = 970 kg/m^3;
Viscosity based on power law; consistency index = 80, Power law index = 0.90,
***Air properties (for the air filled region)*:**
Density, ρ_0 = 1.205 kg/m^3;Viscosity = 1.81e-05 kg/ms
***Body force*:**
Gravitational acceleration g = 9.81 ms^{-2} was imposed in the negative Y direction (see Figures 1 to 3))
Computational Mesh:
12,635 computational cells; typically 1 mm in length
***Star-CD Solver settings*:**
PISO algorithm;
2nd order (MARS) discretization scheme for velocity components and VOF variable,
Iimplicit 1st order temporal discretization scheme, using a time step of Δt = 0.001 s

Table 1- Typical set of data used to run the commercial software in the present study

The following sample results generated by the *STAR-CD,* are typical and closely similar to those obtained using the *FLUENT.*

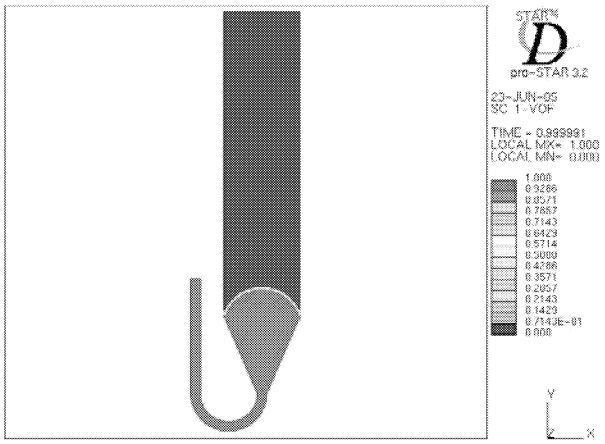


Figure 1- 1 second after the start of moulding process

The study of the filling of the mould cavity, at subsequent time steps, shows an unrealistic pattern in which a gap between the solid walls and the fluid advancing inside the mould is predicted. For example, after nearly 7 seconds the predicted result is as shown in figure 2.

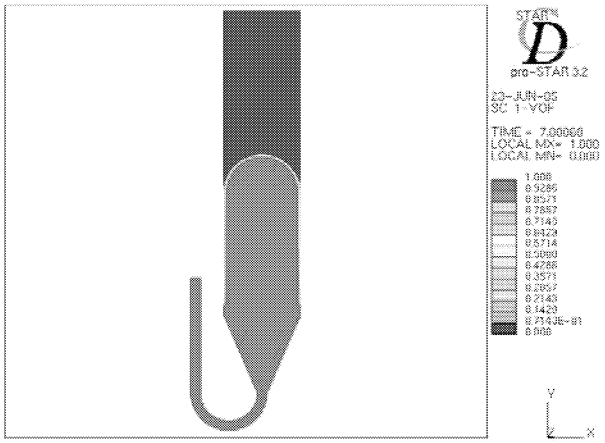


Figure 2- 7 seconds after the start of moulding process

To show the described gap more clearly consider figure 3 in which a part of the mould cavity is enlarged. As there is no physical reason to justify such a prediction it can only be concluded that the imposed wall boundary condition of no-slip is not correct. To examine this concept in this study we have used an alternative approach based on the imposition of partial slip at walls in our in-house developed finite element simulation of the described process. As shown later in this section in this way the artificial gap between the polymer and the mould walls can be eliminated.

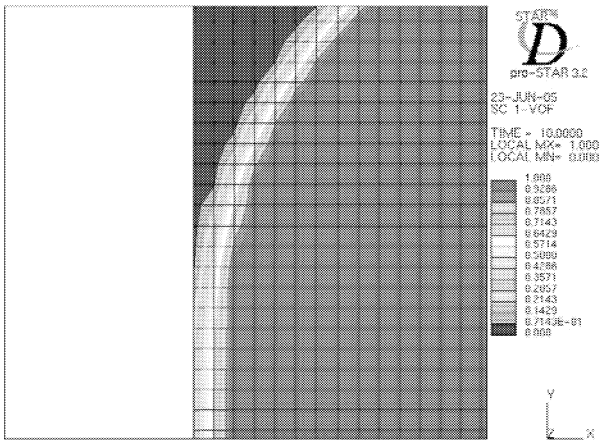


Figure 3- Enlarged section showing the unrealistic gap between the mould wall and thermoplastic fluid

A closer study of the results produced by the described packages reveals other and more subtle differences between these data and the experimental observations. For example, the fronts predicted by these simulations always show near perfect parabolic shapes whilst the experimental data points to somewhat flatter free surface. However a closer examination of the implications of the boundary conditions imposed within Star-CD revealed that the addition of the momentum source term arising from buoyancy need to be explicitly invoked from within the CFD code. By simply invoking ”gravity” from within Star-CD, the required buoyancy related, momentum source term of $s_i = g_i(\rho - \rho_0)$ would not be added to the momentum equation. The change in density which is the cause of the described buoyancy is analogous to the variations of this parameter in a flow domain due to the changes of temperature. Although the mould filling is a very fast and hence mainly isothermal process it seems that this effect cannot be ignored.

The effect of the addition of this momentum source term in creating a flatter polymer advancing front is shown in Figure 4.

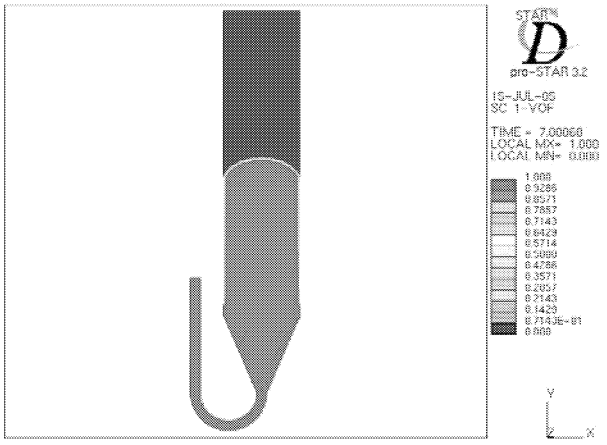


Figure 4- A flatter advancing front as a result of full implementation of body force term due to buoyancy, 7 seconds after the start of moulding process

The comparison of figures 2 and 4 clearly show the flattened shape of the advancing front after the inclusion of the body forces. However, this comparison is qualitative. For a quantitative comparison and to calculate the dependency of flow front curvature on the body forces affecting the mould filling flow one needs to relate the gradient of the curvature (i.e. derivatives of the flow front curve) to the distance from the mould wall. For such detailed analyses to be of value extensive experimental measurements to verify the quantitative, as well as qualitative, accuracy of the simulated geometries are required. Therefore in this paper the presentations are restricted to qualitative comparisons only. Required experimental measurements are currently being carried out by the authors and the results will appear in a forthcoming paper.

The second discrepancy of the original set of results is related to the near perfect symmetry of simulated free surfaces, whereas in reality the free surface becomes asymmetric and slightly tilted towards the outer wall of the mould. To further investigate the underlying phenomena which cause such discrepancies we used our own finite element code. Figure 5 shows the simulated result generated by the finite element scheme, based on the fluid properties and boundary and initial conditions given in table 1 except for the solid wall conditions which is changed from no slip to a very small degree of slip.



Figure 5- Simulated partially filled injection moulding cavity after the imposition of wall slip (using the power law model).

As shown in figure 5, imposition of wall slip resolves the problem of generating already described unrealistic gap between the fluid and the solid walls. However, the predicted free surface still maintains its perfect symmetric parabolic shape. Further investigation reveals the main cause of this deviation from the reality is the elasticity of the thermoplastic. Compressibility has also a very minor effect on making the flow front very slightly flatter. Although the elasticity of thermoplastics is significantly lower than elastomers such as rubber it influences the advancement of the flow front during the injection moulding process. In figure 6 results obtained using the CEF constitutive equation instead of the power law model for the previously described mould filling problem is shown.

The reason to expect an asymmetry in the free surface shape is that as the elastic fluid is subjected to different degrees of deformation on the inner and outer parts of the runner as it flows through this section to fill the mould cavity. The CEF model allows the fluid to reveal this realistic feature affecting an elastic fluid whilst the power law model cannot represent the asymmetry of the normal stresses in the runner section which inevitably has different curvature on its inner and outer walls.

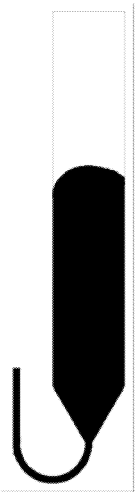


Figure 6- Simulated partially filled injection moulding cavity after the imposition of wall slip (using the CEF model)

Similar to the study of the effects of the inclusion of the body forces the effects of using the CEF model, demonstrated in figure 6, can be quantitatively verified by comparison with experimental measurements.

CONCLUSIONS

The modelling study reported in this paper reveals the comparative importance of various factors in maintaining the accuracy of simulations of injection moulding process of thermoplastics. It shows that effects such as wall slip and elasticity of the fluid have very direct and significant influence on computer model predictions and should never be ignored. Other phenomena such as body forces and slight compressibility of the polymer have secondary influence on the outcome of simulations. However, as the inclusion of these factors in numerical schemes is not complicated, keeping the terms representing such effects in the mathematical models is recommended. This is particularly true for the inclusion of a slight compressibility term in the continuity equation as it improves the stability of the numerical schemes.

ACKNOWLEDEMENTS:
The authors wish to thank the Visteon Corporation for the use of their Star-CD and Fluent licenses during the course of this research work.

REFERENCES

Bird, R.B., Armstrong, R.C. and Hassanger, O. *Dynamics of Polymeric Fluids, Fluid Mechanics*, Vol. 1, Wiley, New York, 1987.

Isayev, A.I.; *Modelling of Polymer Processing.* Hanser Publisher, Munich, 1991.

Ghoreishy, M.H.R. and Nassehi, V.1997 "Modeling the transient flow of rubber compounds in the dispersive section of an internal mixer with slip-stick boundary conditions.", *Advances in Polymar Technology*, 16, 45-68).

Hanspal, N.S., Nassehi, V., and Ghoreishy, M.H.R. 2005. " Numerical analysis of rubber moulding using finite element modelling", *International Polymer Processing*, XX, 238-144.

Nassehi V. *Practical Aspects of Finite Element Modelling of Polymer Processing*, John Wiley & Sons, Ltd, Chichester, West Sussex, 2002.

Nassehi V., and Ghoreishy M. H. R. 1997. " Simulation of Free Surface Flow in Partially Filled Internal Mixers" *International Polymer Processing,*.4, 346-353.

Nassehi, V. and Salemi, R. 1994. "Finite element modelling of non-isothermal viscometric flows in rubber mixing", *International Polymer Processing*, IX, 199-205.

Patankar S.V. *Numerical heat transfer and fluid flow*. Hemisphere, Washington DC, 1980.

Silliman, W.J., and Scriven, L.E., 1980." Separating flow near a static contact line: slip at a wall and shape of a free surface", *Journal of Computational Physics*, 34, 287-313.

Thompson E. 1986. "Use of pseudo-concentrations to follow creeping viscous flows during transient analysis" International Journal of Numerical Methods in fluids, 6, 749-671.

Versteeg H.K.; and Malalasekera W. *An introduction to Computational fluid dynamics: the finite volume method*. Addition Wesley Longman Ltd, Essex, 1995.

Zienkiewicz, O.C. and Wu, J., 1991, "Incompressibility without tears- how to avoid restrictions on mixed formulations", *International Journal of Numerical Methods In Engineering*, 32, 1189-1203

HIGH PERFORMANCE SIMULATION

SIMULATION OF THE EMMIL E-MARKETPLACE MODEL IN SEE-GRID USING THE P-GRADE PORTAL

Peter Kacsuk, Gábor Hermann
Ákos Balasko
MTA SZTAKI
13-17. Kende u. Budapest 1111.
Hungary
kacsuk@sztaki.hu

Lívia Kacsukné Bruckner
International Business School - Budapest
2-4 Tárogató út Budapest 1021
Hungary
lkacsuk@ibs-b.hu

KEYWORDS
B2B e-marketplace, SCM optimisation, grid, SEE-GRID, P-GRADE portal, parameter-study

ABSTRACT

This paper presents a grid-based simulation of a novel e-marketplace model integrated with logistics. The EMMIL model brings together three entity types – seller, buyer and third party logistics service provider – within the same transaction thus enabling combined total cost optimisation for purchase and delivery. The resource allocation in the three-sided auction algorithm is computationally intensive therefore a parallel solution method is needed to make the real time application feasible. The first implementation of the model was carried out for simulation and testing purposes in the SEE-GRID grid infrastructure using the parameter study feature of the P-Grade grid portal. This paper reports about the initial results and shows the simplicity offered by the P-GRADE portal to solve a complex real-life business problem.

INTRODUCTION

Business-to-Business e-commerce has an increasingly important role in Supply Chain Management (SCM) facilitating integration and cost savings. SCM is too complex to set up a "one size fit all" solution. Ongoing researches use different approaches and scenarios, among them e-marketplaces play a substantial goal. Though there are some commercial standards of e-marketplaces, researchers keep seeking for new models in order to improve supply chain efficiency. A good overview of the research directions and tools can be found in (Simchi-Levi et al 2004) Since the exact behaviour of parties in a supply chain model is unpredictable, simulation is a widely used tool in SCM analysis and design. E.g. (Saga et al 2003) describes a concurrent simulation for a procurement e-marketplace integrated with production.
In this paper we report about a grid-based simulation of a novel e-marketplace model called EMMIL (E-Marketplace Model Integrated with Logistics) suggested by one of the authors. (Kacsukné and Cselényi 2005)

The participants of general EMMIL marketplaces are buyers, sellers, and third party logistics providers (3PLs). The resources to be allocated are the following: (1) multi-unit multi-item goods that require physical transportation (2) logistical services: transportation and other optional services like warehousing, labelling, unitising etc. These two kinds of resources are treated separately in traditional e-marketplace models while in EMMIL auction the logistics providers place their offers step by step during the negotiation of buyers and sellers. The marketplace engine combines the proposals and selects the optimal combination. The aim in seller oriented marketplaces is to maximise the profit for the seller, in buyer oriented marketplaces is to minimise the total costs of the buyer.
The complexity of the auction algorithm and the composite bid structure make the optimisation task difficult. The solution that is known today requires lengthy computation that may take several hours to run on a single computer (see example later)
Problems that require computing power beyond a single processor system can be executed on clusters and grid systems. This paper will describe those features and techniques of the EMMIL simulation system that enable its parallel/distributed execution in a user-friendly way.

THE EMMIL PROCUREMENT MARKETPLACE

From now on we focus on buyer-oriented marketplaces since this type is the most elaborated and promising one for a possible real life application. The mathematical foundation of the model is discussed in (Kacsukné 2006) in detail, here only a limited background is given to illustrate the difficulties of the implementation. Participants of the EMMIL procurement marketplace are the following: a single buyer looking for a set of non-digital goods in specified quantities, many sellers (suppliers) who wish to sell certain quantities of some of the required goods and many third party logistics providers that undertake transportation and also other logistical services if needed. The products are assumed to be homogenous from the transportation point of view which means they belong to the same transportation class and are packed uniformly. We assume that unit load bulks will not be broken thus required quantities will be integer numbers.
The marketplace operates a composite reverse auction with discrete rounds of open bidding that alternate between

sellers and 3PLs. In EMMIL 3PLs can bid in a special structure of fix and variable costs. The fix part covers basically the costs of moving the vehicle from starting point (a seller) to the end point (a buyer). The variable cost is proportional to the volume transported where extra costs of handling can be accounted for.

The optimisation problem

In this section we formulate the objective function (1) of the transactions for the EMMIL procurement marketplaces for line-haul transportation without discounts.

$$\min\left(\sum_{k=1}^{M}\left(\sum_{i=1}^{N}P_i^k Q_i^k+\sum_{l=1}^{L}x_k^l\left(F_k^l\left\lceil(\sum_{i=1}^{N}Q_i^k)/Z\right\rceil+V_k^l\sum_{i=1}^{N}Q_i^k\right)\right)\right) \quad (1)$$

Where:

N Number of items (products) to be purchased
M Number of suppliers
L Number of 3PLs
Q_i^k Purchased quantity of product i. from seller k.
P_i^k Unit-price of product i. at seller k.
F^l_k Fix cost of delivering a truckload of goods from seller k.
V^l_k Variable cost of delivering the goods from seller k. (cost/unit load)
Z Standard truckload size
$x_j^l \in \{0,1\}$ decision variable
$x_j^l = 1 \Leftrightarrow$ offer j. of 3PL l. is selected as winner.

Several constraints have to be formulated. First of all, the buyer wants to buy the required quantity of each product as (2) expresses. Constraint (3) says that we select a 3PL for delivery only if we purchase some goods from all the suppliers in that combination. All sellers selected for the transaction – and only those- must be reached by one and only route as (4) formulates. We apply certain business policies as well. Constraint (5) expresses that the number of suppliers should be kept under a specified limit (S) to reduce the additional costs of maintaining relationships with suppliers. (Hohner et al. 2003)

$$\sum_{k=1}^{M} Q_i^k = Q_i \qquad \forall i = 1,2,..N \quad (2)$$

$$x_j^l \le Sign(\sum_{i=1}^{N} Q_i^k) \quad \forall\ k \in \{1,2,..M\} \quad (3)$$

$$\sum_{l=1}^{L} x_k^l = Sign(\sum_{i=1}^{N} Q_i^k) \quad \forall\ k \in \{1,2..M\} \quad (4)$$

$$\sum_{k=1}^{M}\sum_{l=1}^{L} x_k^l \le S \quad (5)$$

We have to determine the x^l_j binary and Q^k_i integer variables to minimise the costs and satisfy the constraints. This problem is a combination of two allocation problems, one for the products and another for the logistical services.

The objective function formulated in (4) is nonlinear and it is not smooth, not even continuous. In order to solve it a linearisation algorithm has been introduced.

The linearisation process

We derive an equivalent problem with a piece-wise linear objective function from (1) by determining the aggregated minimal logistical cost functions for line-haul transportation from each seller as a piecewise linear function. The algorithm is based on (Eso et al. 2001.)
The new objective function is formulated in (6) with the additional constraints given in (6)-(12).

$$Min \sum_{k=1}^{M}\sum_{l=1}^{A_k}\left(\sum_{i=1}^{N} P_i^k Q_i^{kl} + C_k^l y_k^l + V_k^l \sum_{i=1}^{N} Q_i^{kl}\right) \quad (6)$$

$$y_k^l \in \{0,1\} \quad (7)$$

$$Q_i^{kl} \ge 0 \quad \forall i \in \{1,...,N\} \quad (8)$$

$$Q_i^{kl} \le Q_i^{kh} \quad \forall i \in \{1,...,N\} \quad (9)$$

$$\sum_{i=1}^{N} Q_i^{kl} - q_l^k y_k^l \ge 0 \quad \forall l \in \{1,...,A_k\}, k \in \{1,...,M\} \quad (10)$$

$$\sum_{i=1}^{N} Q_i^{kl} - q_{l-1}^k y_k^l \le 0 \quad \forall l \in \{1,...,A_k\}, k \in \{1,...,M\} \quad (11)$$

$$\sum_{k=1}^{M}\sum_{l=1}^{A_k} Q_i^{kl} = Q_i \quad (12)$$

Where

A^k Number of linear pieces of the optimal transportation cost from seller k.
q^l_k quantity at the l. interval at seller k.

The constants in (6) are calculated during the linearisation process as $C_k^l = F_k^l \lceil q_l^k / Z \rceil$

Solution and feasibility problems

The optimisation task (6)-(12) has to be solved for y^l_k binary and Q^{kl}_i integer variables which could be easily done using any standard commercial mixed integer linear programming (MILP) packages – in theory. Our experiences showed that the segmentation resulted in a big model size with a high number of integer variables which led to very long solution times in unpredictable cases. Even in a small size case where we specified 20 kinds of products, 11 sellers and 11 logistics providers it was possible to set the parameters (prices and quantities) so that in unlucky cases we ended up with computation times of 8-10 hours on an up-to date PC. Obviously such a long execution time inhibits the required number of tests as well as the creation of a real auction system.
The remedy for the problem can be the use of parallel and distributed systems like clusters and grid systems. To create a distributed solution we decided to exploit the constraint for the maximum number of sellers. The basic idea was to ssubdivide the MILP problem into a number of similar problems but with only a few variables (determined by constraint S used in (5)) that can be solved in a very

short time and can be done in parallel. The optimal total costs of the different solutions then could be compared and the minimal cost solution could be selected.
To test the distributed algorithm we created a simulator for the EMMIL procurement model with the help of P-GRADE portal developed and maintained by the Laboratory of Parallel and Distributed Systems of MTA SZTAKI.
The aim of the simulation was to determine the feasibility range of this algorithm and determine whether the applicable parameter range was suitable for real life situations.

SEE-GRID AND P-GRADE PORTAL

The EGEE project developed and deployed the LCG-2 (Peris et al. 2005) and gLite (Burke et al. 2007) grid middlewares that are used in all the VOs and grids related to EGEE. One of the EGEE-related grids is SEE-GRID (South-East European Grid) connecting more than 500 CPUs from 31 grid sites of 11 participant countries. SEE-GRID currently supports more than 10 different grid applications among them our EMMIL simulation system.
In order to efficiently develop applications for the grid and run these applications in a user friendly grid environment SEE-GRID adapted P-GRADE portal as its scientific gateway to access the grid. The P-GRADE grid portal is a web based, service rich environment for the development, execution and monitoring of workflows and workflow based parameter studies on various grid platforms. P-GRADE Portal hides low-level grid access mechanisms by high-level graphical interfaces, making even non grid expert users capable of defining and executing distributed applications on multi-institutional computing infrastructures (Kacsuk and Sipos 2005). Although P-GRADE portal is a generic grid application development portal since it is based on the GridSphere portal framework (Novotny 2005), it can be easily adapted for specific application classes and to the needs of specific user communities. More than that P-GRADE portal provides workflow level interoperability (Kacsuk et al. 2006) among different kinds of grid systems like LCG-2, gLite, GT2 and GT4 based grids and as a result users who have got large applications can exploit resources from several grids in parallel. P-GRADE portal is the official portal of SEE-GRID and has been connected to many national (UK NGS, US OSG, US TeraGrid, etc.) and international grids (OGF GIN VO, BalticGrid, EGRID, GILDA, etc.).
To speed up lengthy computations both the cluster and grid solutions have benefits and drawbacks. The cluster solution has the advantage that the overhead of accessing the processors inside a cluster is relatively small. On the other hand its drawback is that the number of available processors is limited by the size of the clusters. The problem is even worse if many users want to use the same cluster. The grid has the advantage that if a cluster is overloaded then another less loaded clusters can be selected from the set of clusters available in the grid. The price for this flexibility is that the initial access time of a cluster is much larger than in the case of a single, dedicated cluster.
Taking into consideration all these aspects we decided to create a solution that can be used on both grids and clusters. We selected SEE-GRID that has many clusters available as grid sites from many countries. In order to gridify the EMMIL system we have been using P-GRADE portal that was extended to support parameter sweep applications in the framework of the SEE-GRID-2 project.
Parameter sweep (PS) applications require the execution of the same code with many different input sets. P-GRADE portal provides a three stage approach for PS applications. In the first stage the so-called generator jobs create the necessary input sets. The second stage is used to run the core application with all the different input sets generated in the first phase. The portal takes care of launching in parallel all the jobs, finding suitable grid resources for them, moving the necessary input files and executables to the selected grid resources and finally, to place the result files of these jobs to a storage element defined by the user. In the third phase collector type jobs written by the user can process the output files generated in the second phase. All the jobs of these phases can be organized into a workflow in a graphical way. As a result users can easily create this type of PS applications without having deep knowledge of grid systems and yet they can automatically exploit as many resources as they need.

THE SIMULATION

The algorithm

Figure 1 shows the application logic of the EMMIL system behind the workflow. The task of the generator job involves creating as many input sets as needed and then the MILP-solver is executed for all sets in parallel on different processors collected from SEE-GRID. The main steps of the algorithms are the following:

1. Generator: Linearisation and subdivision
- Calculate the quantity intervals with the related best 3PL offers (fix costs, variable costs) for all sellers. (This process runs on a single processor because of the short execution time)
- Apply prefilter to eliminate worst sellers, leave best U of them
- Generate all S-ad combination of the best U sellers and pass it to the parameter study input of the core job

2. Core: Solving all MILP problems
- Set up the MILP problem from the input file according to a given formula
- Generate the input data in the format required by the MILP solver program
- Run the MIP program and store the result in a specified grid storage area

3. Collector: Find the absolute optimum
- Compare the total costs, select the best solution and store the results in a specified grid storage area in a specified format.

For MILP solver we applied **lp_solve**, an open source Mixed-Integer Linear Programming system that is free under the GNU license. (Berkelaar et al. 2004.)
Data would be normally collected from participants of the marketplace. In the simulation model we facilitate entering the buyer's data (quantities, basic parameters) individually. For the sellers and 3PLs we generate random data assuming normal distributions where means and standard deviations can be also set as parameters. In order to support the EMMIL simulation system we developed a new input data portlet in P-GRADE portal where the parameters of the bidding data of sellers and 3PLs can be entered in user friendly forms as shown in Figure 2. The output of the simulation has been transported to EXCEL, where we used VBA macros to visualise the results.

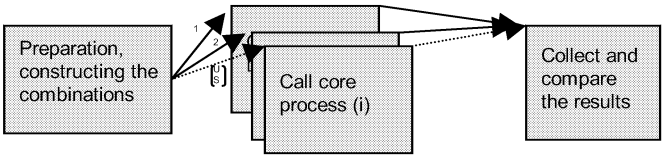


Figure 1: The application logics of the EMMIL workflow

Investigations

The simulated model served two functions. On one hand we wanted to prove the feasibility of running the parallel algorithm on clusters for industrial purposes. On the other hand we studied how the P-GRADE portal and the different grids could cope with this application.

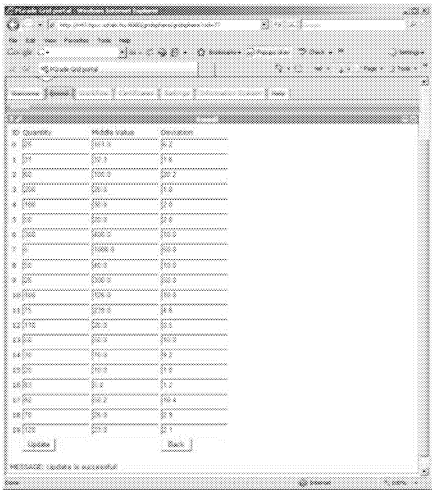


Figure 2: Input form for quantities and price distributions

Experiments have been running in several grid configurations. We started the tests using small numbers for participants first, where the processing times were rather short in the non-distributed case as well. It was not surprising that the execution times up to 10 products, 6 sellers and 6 buyers were normally somewhat longer in the grid than using the non-distributed solution. The advantage of the grid solution became obvious when we started to test cases that ill-behaved in the non-distributed case. One of the worse case scenarios had the following parameters: N=20, M=L=11, S=7 which means 20 products, 11 sellers and 11 logistics providers. The limit for the number of sellers to be selected was 7. Input data for products can be seen in Figure 3. We were particularly interested in a case where we have measured the execution time on a single processor using the original non-distributed algorithm and it was 36400 sec ~ 10 hours. To run the parallel algorithm we have applied a cluster with 10 reserved processors from SEE-GRID. (Measurements of smaller simulation examples in the grid showed that processor reservation is needed in order to achieve real speed-up in the grid.) As a result the execution time has been reduced to 11073 sec ~ 3 hours.
Although this was a significant speed-up, still not good enough for real life situations. We hoped that even better performance could be achieved if we can reduce the grid overhead. In order to reduce the grid overhead we packed up a certain number (k) processes of **lp_solve** calls into a single core job and started to experiment with the granularity. This required only a slight modification in the EMMIL workflow application logics. The advantage of this solution is that by increasing the granularity the number of jobs are reduced k times. We hoped that in this way the proportion of useful computation and grid overhead could be improved. We also wanted to see which value of k is the most optimal and hence we have run three experiments with the following values of k: 5, 10, 33. The results are shown in Figure 4. Parallel time shows the absolute time that was required for the solution from start to the end. Sequential time is the sum of elapsed times on all of the processors used.

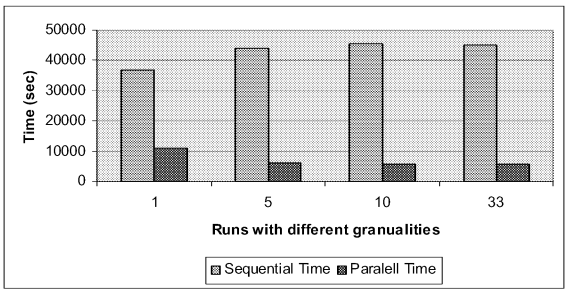


Figure 3: Comparing the Execution Times under Different Granularities

In case of k=5, 10 and 33 the parallel execution time become 5290, 5102 and 5332, respectively. It shows that reducing the number of grid jobs and increasing their granularity halved the execution time and doubled the speed-up of grid execution. The overall speed-up with 10 processors is 7.13 that reduced the original 10 hours execution time to 1.4 hour. Since the lp_solve jobs are independent, reserving more resource in the grid will result in higher speed-up. This is really important when realistic size EMMIL problems to be solved with much more sellers than in this initial experiment.
The experiments also revealed that packaging between k=5 and k=33 has not much impact for the overall speed-up of the application. It means that the grid overhead was

minimized already by the k=5 value. Originally (when k=1) 330 core jobs were generated and with k=5 the number of core jobs was reduced to 66. Therefore in case of k=5 the upper limit for the achievable parallelism and speed-up is 66. If we have 66 reserved processors in the grid we can get 70% of this optimal speed-up (~46) as our grid overhead measures indicated. If we select k=10, we can generate only 33 core jobs that would reduce the achievable parallelism and speed-up to 33. Therefore the optimal value of k is 5 since it reduced the grid overhead as much as it is possible but restricts the achievable parallelism in the least.

CONCLUSIONS AND FUTURE PLANS

What did we learn from this simulation?
First of all it has been proved that the P-GRADE portal is really suitable to quickly develop large parameter study type applications for the grid. The parameter study feature of the portal made possible the first version of the EMMIL simulator to be set up in just a few days. Creating the new specialized input portlet required more efforts since the interconnection of data coming from specialized input portlets with the workflow manager portlet was a non-trivial issue. Solving this problem resulted in a generic method applicable for any other applications to be supported by the portal.
Running the tests was less easy. The first experiments showed that the current version of the gLite middleware used in SEE-GRID (and in many other grids) is not yet prepared for executing hundreds or thousands of jobs if real-time requirements are strong (as in the case of an e-market place application) unless we reserve processors at certain grid sites for the execution. Once the reservation is done the grid environment is feasible to support large applications even with real-time requirements.
Our experiments also proved that the distributed algorithm of EMMIL is working, and it was able to handle cases that were infeasible with the non-parallel solution. However, we are only in the beginning of the road. Testing the simulator with high number of business participants and applying many grid sites is in progress. We are also working on improving the pre-filter algorithm to reduce efficiently the number of jobs generated. Even the initial experiments proved that the EMMIL model would be ultimately feasible if it is supported by a grid execution environment. Experiments with EMMIL also help to improve the SEE-GRID infrastructure and further develop the capabilities of the P-GRADE portal.

ACKNOWLEDGEMENT

The work reported in this paper is supported by the European Commission under contract number 031775 (FP6 SEE-GRID-2 EU Project, www.see-grid.eu)

REFERENCES

Peris A. D. et al 2005. ”LCG-2 User Guide”, Document identifier: CERN-LCG-GDEIS-454439, https://edms.cern.ch/file/454439/2/LCG-2-UserGuide.pdf

Berkelaar, M. Kjell Eikland, Peter Notebaert 2004. ”lpsolve manual” http://lpsolve.sourceforge.net/5.5/

Eso, M. S. Ghosh, J. Kalagnanam, and L. Ladanyi, 2001. “Bid evaluation in procurement auctions with piece-wise linear supply curves”, IBMResearch, Yorktown Heights, NJ, USA, Research Report RC 22219.

Hohner, G., J. Rich, E. Ng, G. Reid, A. J. Davenport, J. R. Kalagnanam, S. H. Lee, and C. An, 2003. “Combinatorial and quantity discount procurement auctions benefit Mars, incorporated and its suppliers” *INTERFACE*S, vol. 33, no. 1, pp. 23Œ35.

Novotny J. 2004. “Developing grid portlets using the GridSphere Portal framework”, http://www.ibm.com/developerworks/grid/library/gr-portlets/

Kacsuk, P. and G. Sipos 2005. “Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal”. In Journal of Grid Computing, Springer, Vol.3. No.3-4, pp.221-238

Kacsuk, P. T. Kiss, G. Sipos 2006. “Solving the Grid Interoperability Problem by P-GRADE Portal at Workflow Level”, Proc. of the Grid-Enabling Legacy Applications and Supporting End Users Workshop, within the framework of the 15th IEEE International Symposium on High Performance Distributed Computing , HPDC’15, Paris, France

Kacsukné Bruckner, L. – Cselényi, J. 2005.: “Principles and Algorithms of EMMIL Marketplaces”, IEEE Conference on E-Commerce München 19.-22. July 2005. 426-430.

Kacsukné Bruckner, L 2006. “Algorithms for E-Marketplaces Integrated with Logistics”, Production Systems and Information Engineering, Volume 3. Miskolc University Press, Miskolc, Hungary 3.-20.

Saga, R. Konzo, T. Aoki, S. Tsuji, H. 2003. “Framework of e-marketplace for concurrent SCM simulator”, 2003. IEEE International Conference on Systems, Man and Cybernetics, Volume: 5, pages: 4785- 4790

Burke S. et al. 2007. gLite 3 User Guide, Document identifier: CERN-LCG-GDEIS-722398, https://edms.cern.ch/file/722398/1.1/gLite-3-UserGuide.pdf

Simchi-Levi, D. Wu, Shen, Eds., 2004. *Handbook of Quantitative Supply Chain Analysis, Modeling in the e-Business Era*, Kluwer Academic Publishers

AUTHOR BIOGRAPHY

PETER KACSUK is the Head of the Laboratory of the Parallel and Distributed Systems in the Computer and Automation Research Institute of the Hungarian Academy of Sciences. He received his MSc and doctorate degrees from the Technical University of Budapest in 1976 and 1984, respectively. He received the kandidat (equivalent to PhD) degree from the Hungarian Academy in 1989.

DECISION SUPPORT METHOD FOR EFFICIENT SEQUENTIAL AND PARALLEL SIMULATION: TIME DECOMPOSITION IN MODIFIED CONCEPTUAL MODELS

László Muka
Elassys Consulting Ltd.
Bég utca 3-5.
H-1026 Budapest, Hungary
e-mail: muka.laszlo@elassys.hu

Gábor Lencse
Department of Telecommunications
Budapest University of Technology and Economics
Magyar tudósok körútja 2.
H-1117 Budapest, Hungary
e-mail: lencse@hit.bme.hu

KEYWORDS

efficiency of simulation, time decomposition, modified conceptual model, resolution of conceptual model, execution time, execution-state model, execution-path model, business process, organisational process, information and communication technology, process simulation, parallel simulation, sequential simulation

ABSTRACT

A new method, the time decomposition in modified conceptual models (MCM) is described. There are introduced two approaches to model the function execution time in MCMs, necessary to decomposition: the execution-state approach and the execution-path approach. There are also described transformations necessary to manage the resolution of MCMs. There is also introduced the concept of preliminary simulation to the time decomposition of MCMs in process simulation environment. There is given an example of using decomposition results to support sequential-parallel simulation decisions increasing the efficiency of simulation.

INTRODUCTION

Efficiency of Simulation Projects

The *efficiency* of simulation projects aimed to support the design of Information and Communication Technology (ICT) and Business Process (BP) systems in an organisation is influenced by some key factors:

These projects usually begin with an *unstructured,* "soft" situation because of the compound system of goals and different roles of the participants (Checkland 1989, Sierhuis and Clancey 2002; Warmerdam and Bredveld 2003).

Common analysis of ICT and BP systems may have advantages but in this case we need to have methods with a *double feature* that is appropriate for both types of systems.
(For example, Warmerdam and Bredveld express a clear need to have methods dealing with the *double feature* when describing the BSM (Business Service Management) method (Warmerdam and Bredveld 2003): by the BSM method the current and future BPs and their requirements (service-level requirements) should be matched with the *performance* and availability of ICT systems required to support them, together with the BPs operating the ICT systems.)

In our previous papers (Muka and Lencse 2006, Muka and Lencse 2007; Lencse and Muka 2007) we outlined a set of methods, the modified conceptual model (MCM) approach, the meta-methodology coping with the unstructured aspect and with the double feature of the simulation projects, outlined above.

In this paper we deal with other key factors of *efficiency*: the problem of finding the right *resolution* of a model and also finding the right *simulation execution methods* (sequential and parallel).

Our proposal to answer these questions is the *method of time decomposition in modified conceptual models,* addressing the problem of *model resolution* and supporting the decision on the use of *sequential and parallel simulation* helping in speeding up the simulation.

CONCEPTUAL MODELS

Traditional Conceptual Models

Soft Systems Methodology (SSM) was developed to be used with unstructured problems, defined by Jackson and Keys (Jackson and Keys 1984, Jackson 1991), in Human Activity Systems (HAS) (Checkland 1989, Checkland and Scholes 1990). The outcome of SSM may also be used with information system analysis and design (Curtis 1989).

The traditional process of SSM consists of seven stages: 1. Entering the Problem Situation, 2. Expressing the Problem Situation, 3. Identifying Relevant Systems, 4. Developing Conceptual Models, 5. Comparing Conceptual with Actual, 6. Defining Changes, 7. Implementing Agreed Changes.

The central idea of SSM is the *conceptual model* of HAS. Conceptual models are the *views of what exist* and not models of what exist in the real world.

The main elements of conceptual models are *key activities* representing *subsystems* of the system. The selected set of subsystems with their logical connections is the conceptual model. The set of conceptual models with defined connections among them form a system of conceptual models. A hierarchy of conceptual models can be found when replacing a first-level conceptual model of a subsystem with its detailed conceptual model.

Modified Conceptual Models

In our previous paper (Muka and Lencse 2006), using ideas of Gregory (Gregory 1993) about necessary and sufficient conditions and also "temporal relations" in conceptual models as the starting approach, we have described the *modified conceptual models (MCM)* with extended harden-up elements to SSM conceptual models, focusing on the design of information systems in an organisation.

In an MCM a key activity is performed generally by an Organisational Process (OP) function or by an ICT system function, that is, any function in the organisation can be performed by some relevant organisational process (P subsystem) with its human resources or by some relevant IT system (IT subsystem) with its technical resources.

Thus MCM elements can be P-type or IT-type depending on what they represent, OP or ICT system function.

An important feature of MCM is that any IT element in the model should be connected to a minimum of one P element in order to have its human resource connection.

To the analysis of necessary and sufficient conditions in MCM, there were three defined element types: F, C and A, that is, PF, PC and PA for processes and ITF, ITC and ITA for IT systems.

F (*function*) is an element performing basic function in the system; element C (*condition*) is providing condition function necessary to perform basic function; while A (*agent*) is an *agent* element ensuring the sufficiency ("motivational" condition) for the basic function to be completed.

Virtual time was also introduced into MCM: *virtual time* is a time sequence assigned to an MCM by giving time labels to its elements.

The virtual time of different MCMs may also be *synchronised* through transient edges (logical connections between MCMs) and condition elements.

In virtual time of MCMs the question of *time decomposition* may also be examined.

TIMING IN MODIFIED CONCEPTUAL MODELS

Execution Time in Modified Conceptual models

Time label T given to an MCM element has the meaning that the event of "function is performed" takes place at T. Extending this definition, time label T may also denote another event: the event of "performance of a function is started" takes place at T.

To the analysis we may also choose the more appropriate definition.

Time label of MCM T_{i-1} denotes on earlier point of time then T_i but there is nothing said about the T_i- T_{i-1} interval.
To give an estimation about time intervals, we introduce *the execution time of key activities (elements)* into MCMs.

Now, we describe timing definitions in conceptual models in a more detailed way:

Timing of function (F) elements
T_i may be the time label of the *finishing event* (the event when the function execution is finished) or time label of the *starting event* (the event when the function execution is started.
If ΔT_i denotes the execution time of a function then in the case of a *finishing event* the execution starts at T_i-ΔT_i,

Timing of condition (C) elements
The main feature of the timing of a condition is that the condition should be ready by the starting point of the execution.
Setup time is the time interval necessary to prepare the condition.
Take-off time is the time interval while the given condition is provided after the function has been executed.
It depends on the system itself whether the condition necessary to be provided during the execution time.
(For example, at a shooting championship it may take some time to set up the target to a shooter and it takes some time to take it off and to set up a new target to a new shooter.)
It depends on the system itself whether the condition should be *held* or not during the function execution. It depends on the system too whether a C element gives a start or finish signal to an F element.

Timing of agent (A) elements
The *agent time* is the time during which the *agent* is used: the agent time in our definition is equal to the execution of a function.
(For example a supervisor at a boxing match should monitor the fight and its conditions during the fighting just to *motivate* the participants to keep to the rules. Before and after the fight monitoring is not necessary but possible.)

Figure 1 shows a summary of the features of the function (subsystem) execution time.

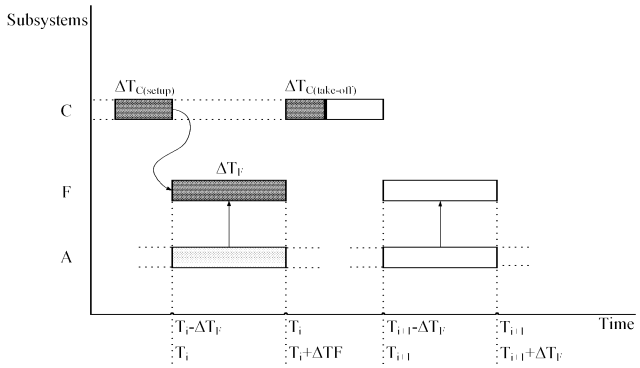


Figure 1 Elements of Function Execution Times

Synchronisation and Execution Time

Different MCMs can be connected through logical connections. These connections are realised by transient edges

going between the MCMs and condition elements of MCMs.

Through transient edges and condition elements *virtual times* of MCMs may also be *synchronised.* (This is the situation in the case of the Customer Help Desk (MCM1) and the Service Department (MCM2) shown in Figure 5. These are separate but cooperating departments.)
Figure 2 shows an example for the execution time in synchronised conceptual models MCM1 and MCM2.
Let us examine the synchronisation of modified conceptual models MCM1 and MCM2: from function element F_{iMCM1} goes a transient edge to C_{jMCM2} and after the execution of F_{jMCM2} it goes back to $C_{(i+k)MCM1}$.

In case of synchronisation of MCMs the setup times of *condition elements* (ΔT_{jSETUP} and $\Delta T_{(i+k)SETUP}$ in Figure 2) used in *synchronisation* behave as execution time being in a *serial order* with function execution times ΔT_i, ΔT_j and $\Delta T_{(i+k)}$.

The other feature of condition elements in synchronisation is that they get initialisation from other function MCM elements and they act as *initialisation* for function elements. (The situation is similar for condition element C_9 in Figure 6.)

These timing features of different condition elements should be taken into account in the model building for time decomposition.

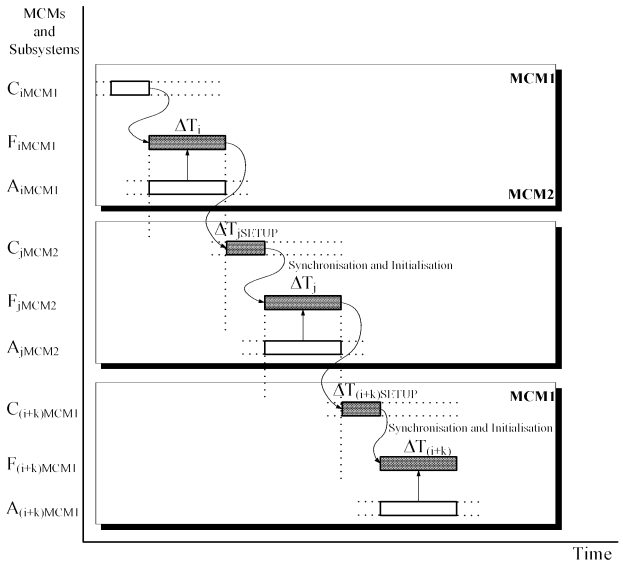


Figure 2 Execution Time in Synchronised Modified Conceptual Models MCM1 and MCM2

RESOLUTION OF MODIFIED CONCEPTUAL MODELS

Resolution of the models is an important factor from the point of view of efficiency of simulation: if the resolution is *too low* the necessary results cannot be reached but if the resolution is *too high* then we loose in modelling work and computing time.
The most *efficient resolution* is a *compromise* between these two requirements.

The appropriate level of resolution may be found by applying a set of transformations to the different parts of the model.

MCM Transformation Methods

Expansion may be used to increase resolution that is to show in greater details in MCM.
Expansion of an IT element gives IT elements showing the internal structure of the original IT element.
Expansion of a P element gives clearly P elements or P and IT elements
Grouping and *integration* are transformations decreasing the resolution of MCM.
Grouping is a transformation treating a group of elements together in modelling and also in simulation.
Integration transformation makes from two or more element one model element.
The main difference between grouping and integration is that grouping will not change the set of model elements, while integration will change it.

SUN Transformation

Now we examine how to increase the resolution of MCMs in the sense of showing different possibilities.
In Figure 3 $C_{2.1}$ – $C_{2.5}$ show the SUN (sufficient but unnecessary) conditions for F_2 as different realisation possibilities of condition C_2.
We suppose that $C_{2.1}$ – $C_{2.5}$ show all the possible realisations of the condition C_2 within the *model limitations*, that is, they are *exhaustive* in the model. This type of a set of exhaustive conditions we call *conceptionally exhaustive set of SUN conditions.*
The conceptionally exhaustive set of SUN conditions uses the limitations derived from the requirements and features of the MCM to set the boundaries in looking for the realisation possibilities of a condition.
It means that instead of purely logical consideration (Gregory 1993) we propose to operate on the requirements and features of MCM and of the system to be modelled too.

Now, we describe an approach of using SUN condition elements in our modified conceptual models.
The possibilities for SUN conditions may be collected in the following steps:

- The interval boundaries for the given condition are defined depending on the model requirements
- If it helps segments (sub-intervals and their boundaries) are defined between boundaries (Figure 3 shows segments SUN Segment1 and SUN Segment2.)
- The realisation possibilities of the condition are listed for the segments
- The set of SUN conditions is checked whether it is *conceptionally* exhaustive

Let us see an example. If we want to by a flat then both the lower and upper limit to find SUN conditions may be given by the flat area: by the accommodation feature in the case of lower limit and by the “too-big-to-clear” feature in the case of the upper limit.

In this case different SUN segments may be the different types of flats: a house with a garden, a flat in a living estate, etc.
The *conceptionally exhaustive* feature of this set of SUN conditions may be from the condition that we decide to use the advertisements only from one internet agency.

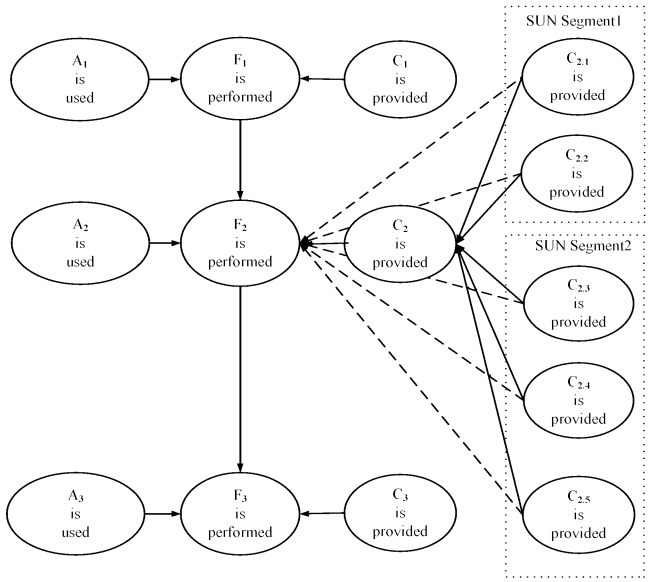


Figure 3 SUN Segments of Condition Elements

If there is an exhaustive set of SUN conditions in the model then at least one member of the set is true. (Dashed line connections in Figure 3 show the "direct influence" of SUN conditions that is the working SUN condition is the condition of F_2.)

Transformations and Timing

Grouping: the timing of the elements defines the timing of the group.
Expansion and integration: the timing of the new elements should be defined first after the situation is the same then in case of grouping.
SUN conditions: the realisation of different SUN conditions may give different execution times. The possibilities of a *conceptionally exhaustive set of SUN conditions* may be examined in serial manner, one-by-one, or in a parallel way as *competing* possibilities.

TIME DECOMPOSITION IN MODIFIED CONCEPTUAL MODELS

To the examination of the time decomposition we use two slightly different approaches:
Execution-state view: the task processing is performed by the changing states defined by the elements of MCMs
Execution-path view: the task travels through the conceptual model using a series of elements of MCMs

Execution-state Model

MCMs show the key activities or subsystems performing key activities of an organisational system.

The execution of a business task initiated by a customer can be shown in MCMs.

A *task* is a sequence of *key activities* initiated by an entering User Request in order to produce an Answer to the User when exiting the MCM.
The *execution-state* is *a set of active subsystems* (F or C) that is subsystems which are taking part in the execution.
When the system is in an execution state the execution of the task is *assigned to this active state*.
During the execution of the task, from the Entry point of time till the Exit point of time there is a series of *changing execution states*, which are necessary to perform the task.
The *transition* among the execution states is performed along the logical connections of the subsystems.
A *time slice* is a time interval during which the execution of a task is assigned to its execution state. Time slices may be different. The time between the Entry and Exit points of MCMs can be shown as a sequence of time slices (Figure 4).

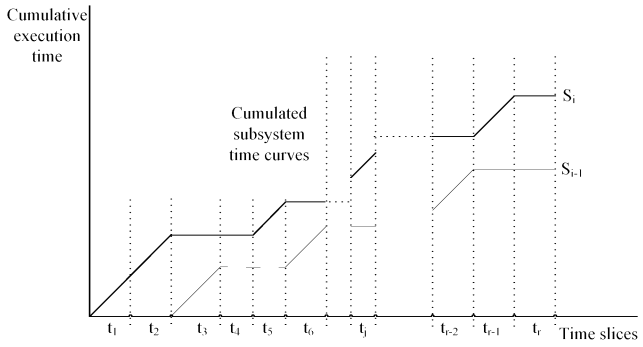


Figure 4 Cumulated Execution Time Assigned to a Subsystem

Figure 4 shows the cumulated time spent by a task in states (subsystems) S_i and S_{i-1} during the execution.

Now, let us express the execution time in formulas:

Let us introduce the *activity variable* a_{ij} with the following meaning:

$$a_{ij}\begin{cases} = 1, \text{ if the task assigned to the subsystem } S_i \ \ (i = 1, 2, \ldots, n) \\ \qquad \text{in time interval } t_j \\ \\ = 0, \text{ otherwise} \end{cases}$$

The time spent in the execution state i may be expressed with the formula:

$$t_{Si} = \sum_{j=1}^{r} a_{ij} t_j \quad \text{where}$$

t_j is the length of time slice j (j=1,2,…,r)
r is the number of time slices
t_{Si} is the time spent in execution state S_i during the task execution.

The full time can be found using the following formulas:

$$t = \sum_{i=1}^{n} \sum_{j=1}^{r} a_{ij} t_j \quad \text{or} \quad t = \sum_{i=1}^{n} t_{Si} \quad \text{where}$$

n is the number of execution states.

The following equation shows the ***decomposition of execution (system) time*** according to the execution state assignment

$$t = t_{S1} + t_{S2} + t_{S3} + \ldots + t_{Sn} = t_{system} \geq t_{execution}.$$

If the execution is assigned to mutually exclusive states that is the set of execution states contains *mutually exclusive elements,* then

$$t_{system} = t_{execution}.$$

If *parallel assignment* is allowed (the states do not exclude each other mutually) the execution time may differ from the sum of time when the subsystems are active. In this case matrix element a_{ij} may be equal to 1 for different values of i at a given j.
Decomposition supports the definition of the critical set of subsystems to be modelled.
The system assignment time may be longer then the Entry-Exit execution time of a task.
Both the $t_{execution}$ and the t_{system} are useful:

- $t_{execution}$ is critical from the point of view of reaction/answer time – that is where and how to decrease these parameters.
- t_{system} is critical from the point of view of resources used by the subsystem – that is how to use a reduced amount of resources.

In a *preliminary simulation* we may take care of the problem of the parallel assignment that is, about the execution time overlapping by just collecting data on parallel assignment too.
In a *measurement,* based on historical data the data on parallel assignment may be missing thus special attention should be paid to the evaluation.

Execution-path Model

The execution path is a route along which the *activating information (message) and/or entity* travel in and between MCMs within the Entry and Exit points of time.

The modified conceptual model can be described as a directed graph MCM(F;C;A;E;TR;ENTRY;EXIT)

where

F, C, A are the set of function, condition and agent nodes respectively (the F, C and A sets may contain P-type or IT-type elements)

E is the set of directed edges of the given MCM

TR is the set of directed transient edges, going to or coming from another MCM, that is transient edges connect elements in different conceptual models

ENTRY, EXIT is the entry and exit edge.
Thus the elements of an execution path may be: F and C elements of MCMs, E edges inside of MCMs, TR edges between the MCMs and ENTRY, EXIT edges to enter and leave the path.
The time is elapsing in nodes while the edges do not cause any delay.
Arrows of E, TR, ENTRY and EXIT edges show the direction of movement along the path.

The sequence of MCM elements which are taking part in the task processing forms an *execution path.*
These elements can be mainly function elements or condition elements in special cases and also logical connections among the elements of MCM.
The execution time of a task depends on the execution time of the function elements taking part in the execution path.

PROCESS SIMULATION APPROACH TO TIME DECOMPOSITION

In the next, we assess the similarities and differences between BPs and MCMs to examine how to apply BP simulation tools to *preliminary simulation* of time decomposition in MCMs, using the principle of *parsimony* (Pidd 1991).

BP and MCM

There are different definitions of a business process:

Processes are structured, measured sets of activities designed to produce a specified output for a particular customer or market (Davenport 1993).
A *business process* is a partially ordered set of Enterprise Activities which can be executed to realise a given objective of an enterprise or a part of an enterprise to achieve some desired end-result (Savén 2002, Koubarakis and Plexousakis 1999).
A *process system* is a set of business processes linked together to perform some Enterprise Function or Subfunction (Lencse and Muka 2006).

From the point of view of the *time decomposition problem,* the MCMs may be treated as a business *process system.*

There are some differences between the process systems and MCMs:

- Key activities, performed by a subsystem of an MCM, may be defined on a higher level then it is usually made in case of processes thus assigning resources for execution should be made differently.
- In the analysis, P-type and IT-type elements (or mixed type elements in case of expanded P elements) can perform the key activity.
- Elements (P and IT too) may be F, C and A elements which play special roles in a system description:
  - F elements (usually) play the role of activity except the feature they do not use resources in a classical way but they do need the presence of conditions C and A.

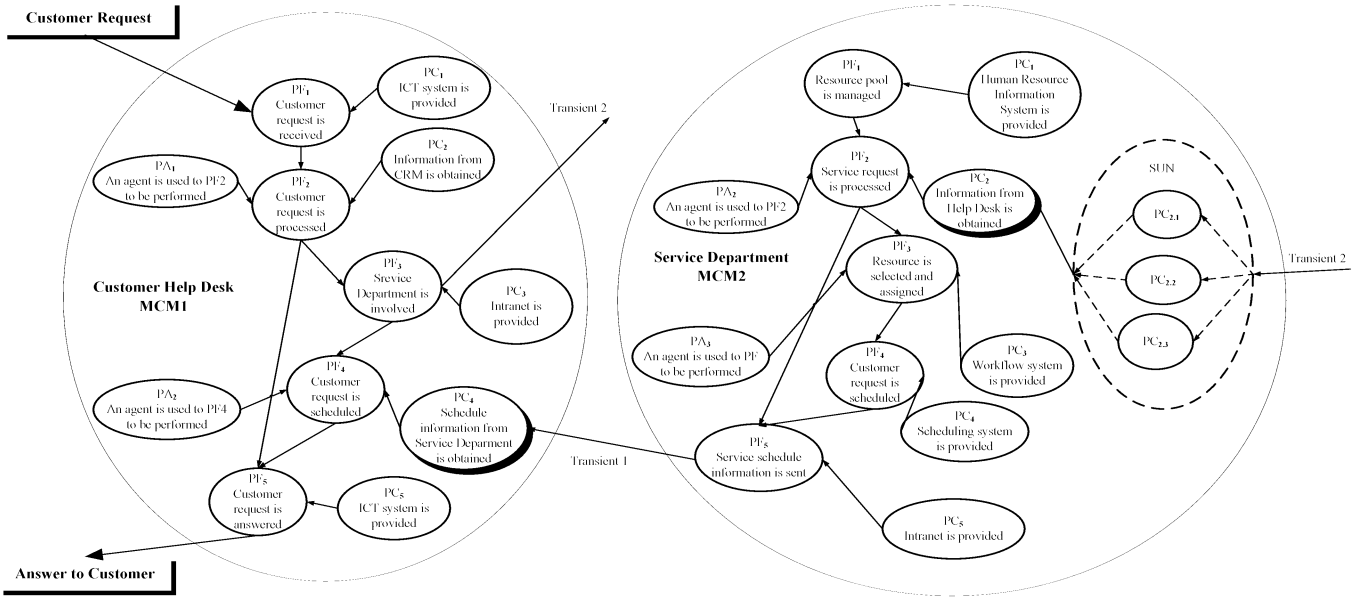


Figure 5 Conceptual Models MCM1 and MCM2 Synchronised Using Condition Elements

- C elements may be treated as "activities providing conditions". The traditional resource conditions of F elements may be connected to C elements.
- In a simple approach, A elements do not take part in time decomposition analysis.

When the execution of a task is in progress the execution of the F activities has an order defined by the routing rules, starting from the first element, the element where the task request enters the conceptual model.

The execution of a C element may become the part of the execution path when an F element has to wait for the condition provided by the C element, or the C element is on the execution path in case of synchronisation.

Process Simulation Elements to Preliminary Simulation of MCMs

Let us examine the elements of Process Simulation (PS) based on the features of *ImiFlow* simulator (Elassys 2007) from the point of view of Preliminary Simulation with MCMs in time decomposition (PSMCM):

PS: The *links* in PS are connections with a direction showing the performance order (time-precedence) of activities.

PS links may be *internal links* (connecting the activities of one process), *external links* (connecting processes forming a process system). The PS links are only logical connections with no capacity limit.

PSMCM: The links in PS are satisfactory to describe the logical connections in and between MCMs during the Preliminary Simulation of MCMs (PSMCM).

PS: *Performing an activity* in PS takes time which is described by a probability distribution (service time profile, activity time-consumption), because many factors are influencing the performance-time. In many cases it is enough to use normal distribution and the expected value of activity time.

T_{Cons}(activity, entity type) Time-Consumption of an Activity – expected value of time necessary to perform the given activity, that is necessary to process the entering entity-type.

PSMCM: The probability tools of activities in PS are adequate to examine for dynamic features of MCMs. Entity type may be used to describe task requests entering MCMs.

PS: The *entity-load* is produced by programmable *entity-generators*, the source of incoming entities. There may be different types of entities entering the process, which are produced by different sources. An entity of a given type has an *arrival profile* which is the arrival time distribution of the entity. The *destinations* of entities are the *exit points* of the process.

PSMCM: The entity-load of PS should model the information/entities traveling through MCMs.

PS: A routing decision may be made using different algorithms: *percentage distribution*, *entity-feature distribution*, *load-balancing distribution.*

There are some other elements influencing the generation and routing of entities:

Fork – Join, and *Split* elements make copies of an entity during the routing process.

PSMCM: We propose to build simple models to PSMCM thus these possibilities are more than satisfactory for this purpose.

PS: *Resource Capacity* has the following useful features:

- ***P_{ARes}(type, month, week, day, time)*** Resource Accessibility – the probability that a resource of the given type is accessible at the given point of time for a given activity
- ***R_n*** Expected value of accessible resource capacity of a given type for an activity ***n***

PSMCM: The resource capacity in the above described classical form should be taken into account as the resource capacity of C elements.

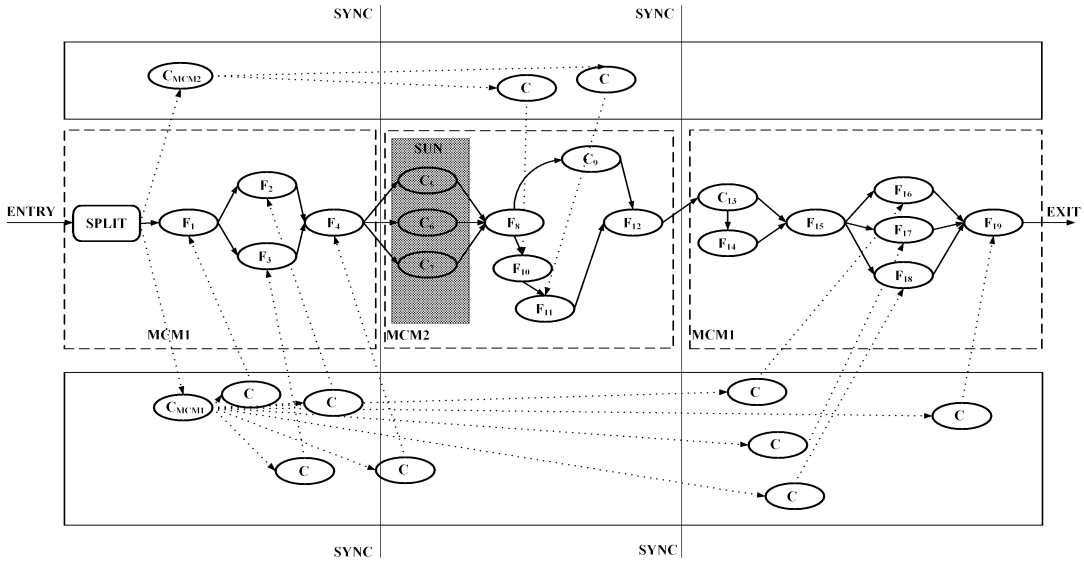


Figure 6 Example of an ENTRY-EXIT Execution Path

Process Simulation Model of an ENTRY-EXIT Path

Let us see an example. Figure 5 shows two cooperating departments, the Customer Help Desk (MCM1) and the Service Department (MCM2) of an enterprise. The Customer Request enters MCM1 through the PF_1 element.
Element PF_2 makes a decision to involve MCM2. The information from MCM1 reaches MCM2 through a set of SUN conditions of PC_2.
PC_2 is the condition "Information from Help Desk is obtained", the set of SUN conditions $PC_{2.1}$-$PC_{2.3}$ may be for example: to get information through e-mail, telephone call or direct database connection. The schedule information leaves MCM2 at PF_5 and returns to MCM1 through PC_4 of MCM1. The Answer to Customer is produced by PF_4 of MCM1.
The ENTRY-EXIT path is the next:
Customer Request- MCM1(PF_1-PF_2-PF_3)-transient edge-
- MCM2(PC_2(SUN:$PC_{2.1}$,$PC_{2.2}$,$PC_{2.3}$)-PF_2-PF_3-PF_4-PF_5)-
-transient edge-MCM1(PC_4-PF_4-PF_5)-Answer to Customer.
In Figure 6 we show the process-model like form of two synchronised systems. (The MCMs of the previous example after some transformations (for example after expanding PF_2, PF_4 elements of MCM1 and PF_2 element of MCM2) may get this structure.)
The ENTRY, through a *SPLIT* element, generates entities for the execution of F_1-F_{19}, SUN(C_5,C_6,C_7), C_9, C_{13} and to set up all the conditions for MCM1 (through C_{MCM1}) and for MCM2 (through C_{MCM2}). All the elements performing activities and their conditions are realised by process activities and by resource capacities connected to the conditions. Fork-Join elements or a percentage distribution may be used at F_1, C_{13} and F_{15}. The SUN element may implement all the three SUN condition possibilities simultaneously in concurrent mode, that is, for example the result produced by the condition element with the shortest execution time may be used each time, thus after the simulation, assessing the results, we may choose the best possibility.

A DECOMPOSITION EXAMPLE

In the case of the execution time decomposition, either we supposed that the execution of the subsystems does not overlap or we restructured the model using transformations to reach this situation in order to be able to approximate the total execution time of a task. However, in a real system the execution of the subsystems may overlap, that is the parallel assignment may be allowed.

Subsystem	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
Execution Time	3	4	2	3	7	6	7	8	5

Table 1 Estimated Execution Times of Subsystems

Figure 7 shows the execution order of the subsystems of an MCM. Table 1 shows the rounded expected value of the execution time of the subsystems in the units of *simulation.*
We have a certain number of processors (in a parallel computer or in a cluster of workstations; let us have 7 processors now) for the execution of our simulation model. *Execution time decomposition* helps us in the assignment of the subsystems to the processors. Figure 7 shows a possibility for the assignment with dotted line. Our goal was that all the processors have approximately equal load.
If we have only 3 processors, we do the assignment according to the continuous line in Figure 7.
Note that besides the execution times, we should also consider the frequency of information exchange of the subsystems during the decision of the assignment.

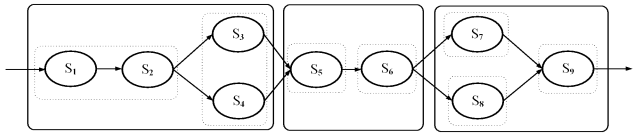


Figure 7 Subsystems of an MCM and Their Assignment to Processors

CONCLUSIONS

In this paper we addressed the efficiency of simulation: we described a time decomposition method in Modified Conceptual Models in order to choose the relevant systems to be modelled, to manage the appropriate model resolution and to support the decision about parallel and sequential simulation. After a brief summary of the essences of conceptual models, we showed a way how timing can be introduced into MCMs.

We described the necessary elements of time decomposition method in MCMs:

- we defined execution time to the elements of MCMs,
- we described transformations necessary to manage the resolution of MCMs,
- we emphasised the importance of appropriate level of resolution of our simulation models and explained how one can achieve it by expansion, grouping, integration and SUN transformations,
- we defined the conceptionally exhaustive SUN conditions as an MCM resolution-increasing transformation, in the way of showing different available possibilities,
- we described the execution-state and the execution-path models of the execution time in MCMs,
- we explained the importance of evaluation of execution time and system time from the point of view of system speed and resources.

In order to use process models to model MCMs, we analysed the similarities between BPs and MCMs and we examined the features of business process simulators as a preliminary simulation tool of the execution time decomposition in MCMs.

We introduced an example of the implementation of the execution-path approach in a process simulation environment.

In the end, we showed an example of using decomposition results to support sequential-parallel simulation decisions increasing the efficiency of simulation.

REFERENCES

Checkland, P. 1989. *Soft systems methodology* In Rational Analysis for a Problematic World, Edited by J. Rosenhead, John Wiley & Sons Ltd

Checkland, P., Scholes, J., 1990. *Soft Systems Methodology in Action* JohnWiley & Sons Ltd., Chichester UK

Curtis, G. 1989. *Business Information Systems* Addison-Wesley, Wokingham, UK.

Davenport, T. H. 1993. *Process innovation: Reengeneering work through information technology* Harvard Business School Press, Boston, Massachusetts

Elassys Consulting Ltd. 2007. *Iminet and Imiflow Systems* http://www.elassys.hu

Gregory, F. 1993. "Cause, Effect, Efficiency and Soft Systems Models" *J. Opl. Res. Soc.* Vol. 44, No. 4.

Jackson, M.C., Keys, P. 1984. "Towards a System of Systems Methodologies" *J. Opl. Res. Soc.* Vol. 35, No. 6.

Jackson, M. C., 1991. *Systems Methodology for the Management Sciences* Plenum Press, New York, London

Koubarakis, M., Plexousakis, D. 1999. "Business process modelling and design – a formal model and methodology" *BT Technol. J.* Vol. 17, No. 4.

Lencse, G., Muka, L. 2006. "Expanded Scope of Traffic-Flow Analysis: Entity Flow-Phase Analysis for Rapid Performance Evaluation of Enterprise Process Systems" *Proceedings of the 2006 European Simulation and Modelling Conference* (ESM'2006) (Toulouse, France, 2006. Oct. 23-25.) EUROSIS-ETI, 94-98.

Lencse, G., Muka, L. 2007. "Combination and Interworking of Four Modelling Methods for Infocommunications and Business Process Modelling" *Proceedings of the 5th Industrial Simulation Conference* (ISC'2007) (Delft, The Netherlands, 2007. Jun. 11-13.) EUROSIS-ETI, 350-354.

Muka, L., Lencse, G., 2006. "Developing a Meta-Methodology Supporting the Application of Parallel Simulation" *Proceedings of the 2006 European Simulation and Modelling Conference* (ESM'2006) (Toulouse, France, 2006. Oct. 23-25.) EUROSIS-ETI, 117-121.

Muka, L., Lencse, G. 2007. "Hard and Soft Approaches in a Simulation Meta-methodology" *Proceedings of the 5th Industrial Simulation Conference* (ISC'2007) (Delft, The Netherlands, 2007. Jun. 11-13.) EUROSIS-ETI, 17-22.

Pidd, M. 1991. "Computer simulation methods" In *Operations Research in Management*, Edited by Littlechild, S., and Shutler. M., Prentice Hall, UK.

Savén, R. 2002. "Process Modelling for Enterprise Integration: review and framework" Department of Production Economics, Linköping Institute of Technology, Linköping, Sweden

Sierhuis, M., Clancey, W. J. 2002. "Modeling and Simulating Work Practice: A Method for Work System Design" *IEEE Intelligent Systems* Vol. 17, No. 5. pp. 32-41.

Warmerdam, M., Bredveld, P. 2003. *A Holistic Approach to Delivering the Value of IT: Business Service Management* IDC White Paper, Framingham USA, http://www.idc.com

BIOGRAPHIES

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994 and his Ph.D. in 2001. The area of his research is (parallel) discrete-event simulation methodology. He is interested in the acceleration of the simulation of communication systems. Since 1997, he works for the Széchenyi István University in Győr. He teaches computer networks and networking protocols. Now, he is an Associate Professor. He is a founding member of the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University.

Dr Lencse does R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998.

He works part time at the Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005. There he teaches computer architectures.

LÁSZLÓ MUKA graduated in electrical engineering at the Technical University of Lvov in 1976. He got his special engineering degree in digital electronics at the Technical University of Budapest in 1981, and became a university level doctor in architectures of CAD systems in 1987. Dr Muka finished an MBA at Brunel University of London in 1996. Since 1996 he has been working in the area of simulation modelling of telecommunication systems, including human subsystems.

Dr. Muka is a regular invited lecturer in the topics of application of computer simulation for performance analysis of telecommunication systems at the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University of Győr.

CASE STUDY: PERFORMANCE EVALUATION FOR COMMUNICATION STRATEGIES IN DISTRIBUTED EMBEDDED SYSTEMS

Falk Berger, Bernd Däne, Alexander Pacholik, Wolfgang Fengler
Department of Computer Architecture
Ilmenau Technical University
98684 Ilmenau, Germany
{falk.berger|bernd.daene|alexander.pacholik|wolfgang.fengler}@tu-ilmenau.de

KEYWORDS

Signal processing, Computer networks, Discrete event simulation, Performance analysis, Queueing

ABSTRACT

To analyze common topologies of communication systems there are several probabilistic analysis strategies existing. Regarding the analysis of the influence of routing strategies probabilistic assumption is hard to validate. We used a discrete event simulation based analysis method to examine the influence of routing strategies within communication structures using a ring topology. By modeling the routing strategy in detail we are able to easily determine a realistic performance behavior under different load conditions.

INTRODUCTION

With the growing demand of high performance computing in embedded devices, there is more and more demand for devices with more than one computing resource. In modern systems these cpus are connected together via point-to-point connections like HyperTransport (Troden and Anderson 2003), PCIe or serial RapidIO. These multiple devices could be connected in different architectures, which all have there own strengths and weaknesses in properties like performance, price or scalability. Because we have searched for a cheap and easily scalable solution for our computing system we have choosen an unidirectional ring architecture. Every node in an unidirectional ring structure is only connected to its two neighbors, so only two links per node are required. To unload the cpus from the routing task, we introduce a tunnel device (Troden and Anderson 2003) for every communication node as shown in Fig. 1.
The tunnel device handles all routing tasks and has an incoming and an outgoing link to the neighbors and one link to the connected computing resource. There are two possible data sources for a tunnel device, so there must be a strategy to handle the case when there is data to be sent from both of these sources. To examine the influence of the routing strategy on the performance of the system, we modeled the system using a discrete event simulation. A stochastic analysis method like markov chains would be insufficient in this case, because the routing strategies depend on the correct order of data packets. So we have to model the system to get some realistic performance data.

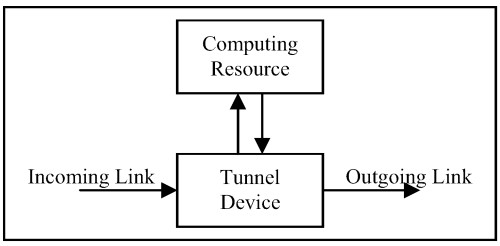


Figure 1. Tunnel Device

The purpose of this paper is to show how we used a model of a communication system to analyze it in terms of performance. At first we will give a short overview of the modeled system. Next we will give a more detailed description of a tunnel device. Design and Properties of the model will be described next. This will be followed by a description of the examined routing strategies. Then we will give some simulation results along with a discussion of it. And last we will give some conclusions and an outlook for further work.

MODELED SYSTEM AND TOOLS USED

This paper is motivated by an embedded system (Däne and Berger 2004), in which some DSPs (digital signal processors of type Texas Instruments TMS320C6713) (Castille 1999) should be connected together in a circular structure to avoid possible bus bottlenecks. This is somehow similar to the ring structure from the Scalable Coherent Interface (SCI) standard (IEEE 1596-1992) (IEEE Std. 1993). Altera Cyclone II FPGAs (Field Programmable Gate Arrays) are used as tunnel devices. As communication link between the tunnel devices, an LVDS-(Low Voltage Differential Signaling) connection with 8 data lanes and 1 clock lane, running at a connection speed of 600 Mbit/s per lane is used. This results in an overall connection speed of 600 MByte/s. The FPGAs are connected to the DSPs through a memory interface with a connection speed of 400 MByte/s. Every node runs in his own clock domain because the clocks are derived from the DSP node. Because of this, the tunnel device must handle these asynchronous clock domains. Fig.

2 shows a graphical representation of the communication infrastructure of the modeled system.

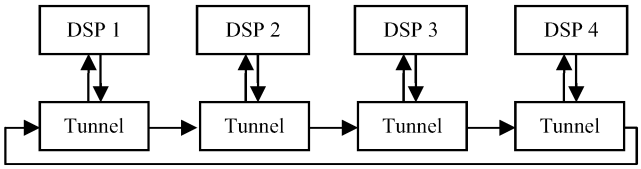


Figure 2. The Modeled System

The tunnel devices shall be implemented in Cyclone II FPGAs from Altera (Altera Corp. 2007). A simple schematic overview of the internal structure is shown in Fig. 3. Serial data received from the eight LVDS-lanes is decoded and deserialized by the LVDS-Receiver. The data packet is then examined by the switch and if the data is for the connected DSP then it is put to the DSP interface. Otherwise it is put in the ring FIFO. Data packets coming from the DSP are put into the node FIFO. Now it is the turn of the FIFO manager to decide from which FIFO to get the next data packet. The LVDS-Transmitter then serializes and encodes the data.

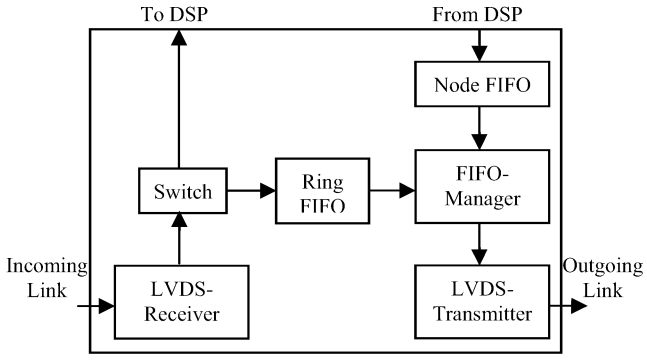


Figure 3. Schematic of a Tunnel Device

The Model

The model of the examined system is a cycle accurate simulation. This means that the model can simulate more than just the data packet transfer and routing strategies. It also can handle all low level tasks needed for initialization and clock deskewing. Fig. 4 shows a top level view of the model. The model consists of four communication nodes. Every node consists of a low level link, a high level and a data source model. The nodes are connected together via a delay primitive to simulate the wire delay. Deeper levels of the model consist of several modules that are not shown here because of complexity.

The low level link model handles low level tasks of the communication protocol, as the initialization, the insertion of idle packets or the deskewing of the incoming clock. The high level model handles most importantly the routing of the data packets and secondly the disassembly of the data packets in single data items. The reason for the separation of the communication node in a low and a high level module is that we only need to replace the high level module to examine another routing strategy.

The data sources generate new data packets at a random basis. Packet sizes as well as destination nodes are randomized, besides the time distribution of the packets. This can be extended to a more complex data source where new data packets are generated in reaction to external events or to other incoming packets.

Tool Used

The modeling tool used in this paper was *MLDesigner* of *MLDesign Technologies Inc.* (Zerbe et al. 1999; Zerbe 2003). This tool provides multi-domain modeling with graphical block models and other graphical means. It extends the well known tool *Ptolemy Classic* from University of Berkeley (Buck et al. 1994). Compared to *Ptolemy Classic* the tool *MLDesigner* provides extended domains and libraries, more effective simulation procedures and a new user interface.

Examined Scheduling Strategies

Now we will introduce the examined Scheduling Strategies. One mean result of the modeling process will be the comparison of effectiveness of these strategies. For all strategies there is one restriction: data packets are not interruptible. If a node has started a transmission of a packet, the next packet can only be started after the actual one is finished. To get a reasonable worst case latency, data packets have a limited length of 16 data entries. Not covered in this paper is the bandwidth reservation technique, because it uses a very static scheduling method.

Roundabout Strategy.

This is one of the simplest scheduling strategies. Data packets in the ring-FIFO always have priority over the ones from the node-FIFO. This is named after the policy used in a roundabout traffic system. The SCI standard uses this strategy. Data Packets could be blocked for a longer time if the incoming link has a 100% load and all data packets are routed through this node to the next.

Timeslot Strategy.

In this strategy every node is allowed to insert in data packet from the node-FIFO after every n^{th} (in this model every 4^{th}) packet from the ring-FIFO. In opposite to a pure timeslot strategie a node is also allowed to insert own packets if the ring-FIFO is empty. Data packets couldn't be blocked for a longer time.

First Come, First Serve.

Another simple strategy. Every incoming data packet is simply put into one global FIFO instedt of the two FIFOs from the first two strategies. No data packets could be blocked for a longer time because the first packet who enters a FIFO leaves it first.

Longest Path First.

A little bit more expensive strategy, where data packets gets a priority according to the nodes it must pass to get to the

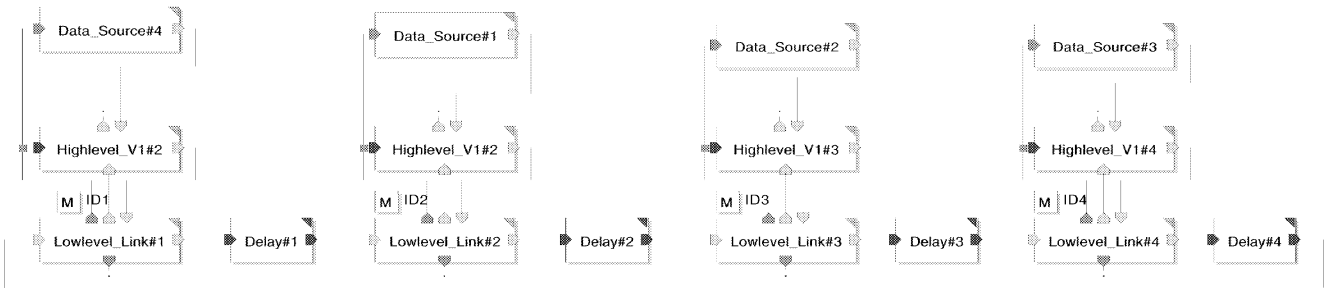


Figure 4. MLDesigner Diagram Of The Modeled System

destination. Because of the priorization of the data packets, blocking for a longer time could happen for lower priorized data packets.

Most Data First.
In this strategy the data is pulled from the FIFO, which has the most data available. Blocking for longer time is possible in the same case as in the roundabout strategy, but only if the node has very little number of data packets from the DSP.
All these strategies have been implemented by several versions of "high-level" modul.

SIMULATION RESULTS

To simulate the model the data sources (Fig. 4) can generate data packets with different lengths, target node and different parameters for the probability distribution for the delay between packets. For this paper we used a model where every node generates a similar amount of data packets. So we could examine the system under different load conditions of the links. As system load we mean the average utilization of the links.
Latencies of messages have been measured in the model by observation elements. The simulated time for a single simulation runs was 150 µs.

50% System Load

For the first examination we used a simulated load of 50 percent. The resulting average values and standard deviation of the latency is shown in Table 1.

Table 1. Latency With 50% System Load

Strategy	Average (in ns)	Stddev (in ns)	Maximum (in ns)
Roundabout	129.43	56.54	375
Timeslot	134.68	55.10	408
First Come, First Serve	129.37	56.29	375
Longest Path First	129.26	55.38	355
Most Data First	132.69	57.38	382

As there can be seen, there are no significant differences for the different strategies for such an average load of the system.

95% System Load

To see more different results for the strategies we use this time a heavier load of 95%. The results in numbers for this examination are shown in Table 2. Total packet count was about 2000 packets per node.

Table 2. Latency With 95% System Load

Strategy	Average (in ns)	Stddev (in ns)	Maximum (in ns)
Roundabout	473.56	368.03	3408
Timeslot	508.53	309.60	2082
First Come, First Serve	488.19	272.21	1815
Longest Path First	469.96	353.87	5922
Most Data First	501.91	278.28	2135

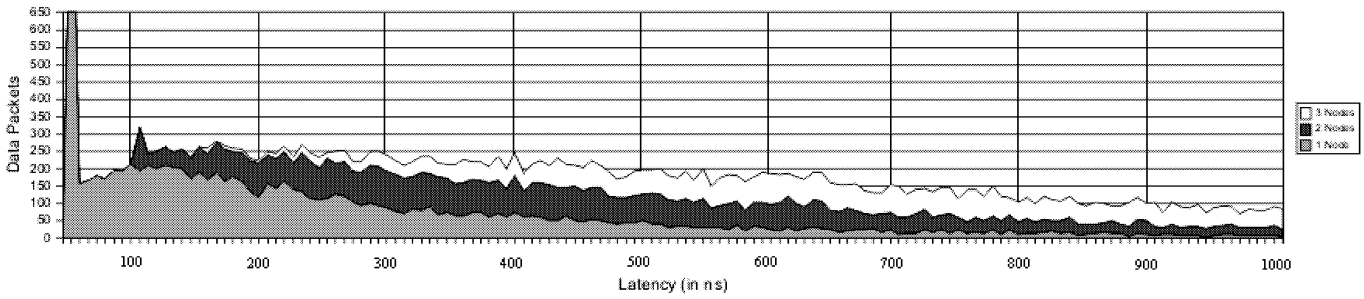


Figure 5. Latency Distribution For The „First Come, First Serve"-Strategy Under 95% Load

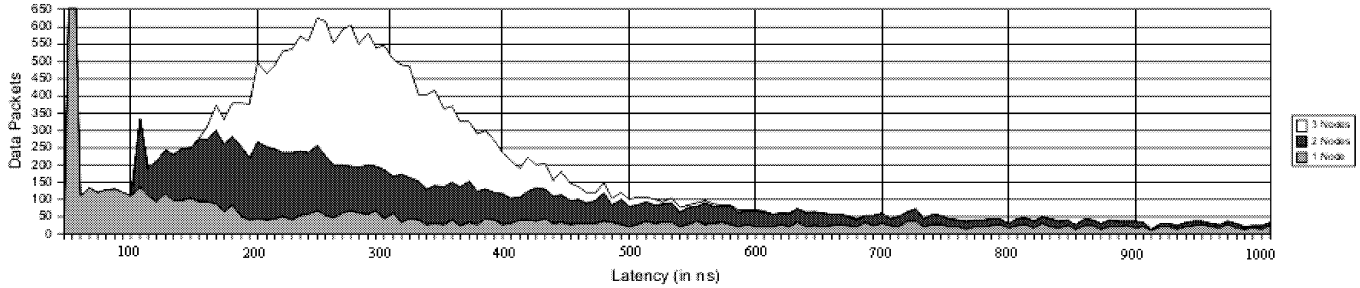


Figure 6. Latency Distribution For The „Longest Path First“-Strategy Under 95% Load

Under such a heavy load condition the strategies show more differences. Especially the maximum value of the latency shows strong differences. It shows that the strategies which do not block out special data packets for a longer time (timeslot and first come, first serve) have the lowest maximum latencies.

Fig. 5 and Fig. 6 show the latency/data packets diagrams for the “First Come, First Serve” and the “Longest Path First”-strategy, where different travel distances are marked in different colors. As there can be seen at the beginning of the latency-axis there are first only the data packets which travel only to the next node. Packets which travel over more nodes need more time. Packets which travels only one node need at least 50 ns, for two nodes 100 ns are needed and 150 ns are needed for three nodes.

For the “First Come, First Serve”-strategy there can be seen that data packets with a travel distance over three nodes are only increasing slowly. This is an expected behavior, because these packets must queue up newly on every node.

A special observation could be made for the “Longest Path First”-strategy. As can be seen, due to the prioritization of data packets with longer travel distances, all packets which travel over three nodes are delivered within 540 ns. But now, data packets which travels only to the next node needs significantly more time, nearly 6 µs in our case, which is by far more than the other strategies.

CONCLUSIONS

If there is only a medium loaded communication system, then it nearly doesn’t matter which strategy is chosen for picking the next data packet for transmitting. Under heavy load it plays a mayor role. Surprisingly not the “Roundabout”-strategy is the best strategy concerning the maximal possible latency under heavy load conditions. Surprisingly in that, that it is used in many communication standards.

Without further investigations in other scheduling strategies and simulation setups, it seems that the “First Come, First Serve”-strategy is best suited for such communication problems. Also this strategy is very easy to implement and makes very good use of memory resources, because all data packets are stored in one FIFO and no overhead like a priority flag is needed.

If you don’t care about the long maximum latency of the “Roundabout” or the “Longest Path First”-strategy, then one of this may be your strategy of choice. With a little bit more relaxation of the strong prioritization of the “Longest Path First”-strategy the long maximum latency can possibly be reduced.

In general it has been shown that the model based approach led to useful results and can be helpful when designing the actual hardware/software system.

FURTHER WORK

One of the next tasks is to replace the random data source by a more complex one. Also we want to examine further routing strategies. To have a good comparison for the strategies we want to develop an “ideal”-strategy which has access to data which the normal routing algorithms do not have, especially the timestamp of the generation of the data and a projected arrival time of the packet. And at last we want to implement the best strategy in hardware.

ACKNOWLEDGMENT

This work has been supported by *Deutsche Forschungs-gemeinschaft* (German Research Foundation) under grant *SFB 622*.

MLDesigner © 2005 MLDesign Technologies, Inc. All rights reserved. *www.mldesigner.com*

REFERENCES

Altera Corp. 2007. *Cyclone II Device Handbook*. San Jose, USA.

Buck, J.; S. Ha; E.A. Lee; and D.G. Messerschmitt. 1994. “Ptolemy. A Framework for Simulating and Prototyping Heterogeneous Systems“. *Int. Journal of Computer Simulation*, 22(4), 155-182.

Castille, K. 1999. *TMS320C6000 EMIF to External SBSRAM Interface.* Application Report SPRA533, Texas Instruments Inc., Dallas, USA.

Däne, B. and F. Berger. 2004. “A Multiprocessor DSP System for a High Throughput Control Application”. In *EDERS 2004: The European DSP Education and Research Symposium. Birmingham*, UK.

IEEE Std. 1596-1992. 1993. *IEEE Standard for Scalable Coherent Interface (SCI)*. The Institute of Electrical and Electronics Engineers, Inc.

Troden, J. and D. Anderson. 2003., *HyperTransport System Architecture* Addison-Wesley Professional, ISBN 978-0321168450.

Zerbe, V.; U.Freund; and H. Salzwedel. 1999. “Mission Level Design of Control Systems“. In *SCI/ISAS’99 Multiconference on Systemics, Cybernetics, Cybernetics, Informatic*s, vol. 7, Orlando, USA, 237-243.

Zerbe, V. 2003. “Mission Level Design of Complex Autonomous Systems”. In *ETRAN 2003 IEEE Conference Proceedings*, Herceg Novi, Serbia, 55-59.

OPTIMIZING THE TIME WARP PROTOCOL WITH LEARNING AUTOMATA

Jun Wang and Carl Tropper
School of Computer Science
McGill University
Montreal, Quebec, Canada
Email: jwang90, carl@cs.mcgill.ca

KEYWORDS
Time Warp, Learning, Learning Automata, Dynamic modelling, AI in simulation, VLSI

ABSTRACT

In many existing adaptive Time Warp protocols, the control mechanism adjusts the control parameters based on a pre-determined analytical model of the system, expressed as a mapping from system state to the control parameter. The underlying assumption is that the mapping is optimal. In this paper we present a different approach that is based on the theory of learning automata. In contrast to the analytical model-based methods, using learning automata, we assume no knowledge of the underlying Time Warp system, and attempt to identify the optimal control parameter through experience only. Our learning automata-based techniques were implemented in a distributed VLSI simulator with the objective of finding the optimal size of a bounded time window. Our experiments using two benchmark circuits indicated that it was successful in doing so.

INTRODUCTION

In parallel discrete event simulation (PDES), logical processes (LP) that work in parallel to carry out the simulation task must coordinate in order to ensure simulation correctness. Over the years, two major classes of synchronization protocols have been developed for distributed simulationFujimoto (1997). In a conservative protocol, an LP executes the next scheduled event only when it is safe to do so. Waiting for safety assurance means that the LPs have to either block for some period of time or send synchronization information to each other, thereby incurring extra overhead. The most well-known optimistic protocol, Time Warp, on the other hand, always assumes it is safe to execute the next scheduled event. Whenever the assumption proves to be incorrect by the arrival of an out-of-order external event, the LP must undo all computations subsequent to the mistake and *roll back* to a previous state. While Time Warp has the potential of making better use of the system's parallelism, it is well known that over-optimism can lead to instability. In the worst case, the LPs spend most of their time rolling one another back, making it impossible for the simulation to progressLubachevsky et al. (1991).

From the efforts to address the problem of over-optimism a class of hybrid protocols have emerged, which either impart optimism to conservative protocols or limit the optimism of optimistic protocols Das (2000). In this paper we are only concerned with adaptive Time Warp. As defined in Reynolds (1988), an adaptive protocol modifies its behavior dynamically in response to changes in the simulation process. In other words, an adaptive protocol dynamically adjusts certain control parameter(s) of the simulation in order to achieve the best performance possible. In adaptive Time Warp the control parameter is usually based on an analytic model and is expressed as a closed-form function of system state measurements. The goal of the adaptive protocol is usually to reduce excessive rollbacks. The underlying assumption is that this mapping from system state to control parameter is optimal.

In this paper, we present an approach to determining the optimal control parameter(s) for a Time Warp simulator that is based on techniques of learning automata, a branch of the theory of optimal control. The most important factor that makes learning automata an attractive method to optimizing Time Warp is the fact that it requires no analytic model of the system. The control parameter is not a pre-defined function of simulation state. Rather, it is determined from simulation experience. In other words, instead of assuming an optimal control strategy, we attempt to acquire one through learning. In addition, the performance of learning automata is based on solid mathematical foundation.

The rest of the paper is organized as follows. Section 2 provides background information on adaptive Time Warp. Section 3 gives an overview of learning automata. In section 4 we discuss how we can use learning automata techniques in a specific optimization problem of the Time Warp protocol. In section 5 we present test results for our distributed VLSI simulator on some benchmark circuits and our analysis of the results. Finally, section 6 contains our conclusions.

ADAPTIVE TIME WARP

It is well known that Time Warp is prone to an excessive usage of memory and an explosive growth in the number of rollbacks. Excessive memory usage can cause performance degradation in virtual memory systems, and uncontrolled rollbacks can become dominant in the simulation process and even cause system deadlock in the most extreme cases.

An obvious solution to this problem is to limit optimism. For example, an early protocol called Moving Time Window (MTW)Sokol et al. (1988) controls optimism by only allowing events whose timestamps are within a certain window to be executed optimistically. The window size, however, is determined in advance and remains constant throughout the simulation. Obviously, the difficulty is to statically determine an appropriate window size before the simulation.

Adaptive Time Warp protocols go one step further. Instead of using a pre-determined constant value for the control parameter(s), they adjust the control parameters dynamically based on "knowledge of selected aspects of the state of the simulation"Reynolds (1988). Not surprisingly, most adaptive protocols choose to control memory usage or use a time window which blocks overly optimistic executions. For example, in Panesar and Fujimoto (1997) optimism is controlled by adjusting the size of a window that defines an upper bound on the number of uncommitted events an LP can schedule; in Palaniswamy and Wilsey (1993) a moving bounded time window is defined such that only events scheduled within the window can be optimistically executed. DasDas (2000) provides a survey of the adaptive Time Warp protocols.

While the adaptive protocols have all been shown to outperform pure Time Warp in some applications, we feel that there are some drawback to their use.

The first problem is the use of analytic models for the control parameters. The value of a control parameter c is expressed as a closed form function of a vector $\vec{S}$ of some system state variables using a pre-determined model: $c = f(\vec{S})$. For example, the window in Panesar and Fujimoto (1997) is based on a queuing model; the checkpoint interval in Lin et al. (1993) is defined as a function of parameters such as state-saving time and event execution time. Obviously, the quality of the protocols is heavily dependent on the quality of the model. Because of the simplifications which are necessary to make in the course of developing such a model, the control strategy is not certain to be optimal. (A notable exception to using an analytic model is the adaptive protocol proposed by Ferscha Ferscha (1995), where the control element of the system tries to build a statistical model of the system from simulation experience.)

Another drawback, and probably the most common one, is the lack of evaluation of the effectiveness of the control mechanism. The control function is assumed to be optimal. A typical example is an adaptive protocol that tries to control rollbacks by forcing an LP to block for a short period of time based on its current local time and the global virtual time (GVT). The protocol calculates an amount of blocking time at an LP using a pre-defined function and blocks the LP in hope of controlling rollbacks. It never measures whether the blocking is effective. Worse yet, there is the danger of pursuing subgoals such as reducing the number of rollbacks while ignoring the real goal, which is to minimize simulation time.

The learning automata-based techniques we present in this paper do not require a pre-determined model. In fact, We do not assume any knowledge of what value the control parameter should assume in any given system state, because that is exactly the knowledge we try to acquire. In other words, we try to optimize the system performance p as a function of the control parameter c and system state $\vec{S}$ without the closed form of a function.

LEARNING AUTOMATA

This section gives an overview of the theory of learning automata. For a much more detailed introduction, the reader is referred to Narendra and Thathachar (1989).

Introduction

The process learning has been extensively studied for a long time in various scientific disciplines. The automata approach to learning structures the learning agent as an automaton. Like other automata, a learning automaton is defined by a finite set of states, a set of events or inputs that may cause the automaton to transit from one state to another, and a set of outputs when the state transitions occur. In each state, the learning automaton has a finite set of actions from which it can choose one to perform on its environment. As the result of the action, the environment provides a favorable or unfavorable response with a probability that is unknown to the automaton. The objective of the learning automaton is to determine for each state an optimal action such that the probability of receiving favorable responses from the environment is maximized.

More formally, a learning automaton is defined as a five-tuple $\{\Phi, X, Y, F, H\}$, where the elements represent the following entities.

- The state set $\Phi = \{\phi_1, \phi_2, \ldots, \phi_s\}$ is the finite set of the automaton's possible states. The state at the instant n is denoted by $\phi(n)$.
- The set X is the possible values of the automaton's inputs. The set can be finite or infinite. The input at instant n is denoted by $x(n)$. The inputs of the automaton are the outputs of the environment.
- The set $Y = \{y_1, y_2, \ldots, y_r\}$ is the finite set of the

automaton's possible outputs, or actions. Since the automaton and its environment are connected in a feedback manner, the automaton's outputs are inputs to the environment. The output at instant n is denoted by $y(n)$.

- The transition function $F : \Phi \times X \to \Phi$ determines how the automaton transits from one state to another under certain inputs. The function can be deterministic or stochastic.
- The output function $G : \Phi \to Y$ determines the output of the automaton in terms of its state. The function can be deterministic or stochastic.

In the following, we shall confine our discussion to the so-called P-model environment, where the input set X consists of only two values: 0 and 1 for favorable and unfavorable responses, respectively. Furthermore, we shall only consider stationary environments, where the penalty probability of each action does not change with time.
One important concept concerning the performance of a learning automaton is the average penalty. For a stationary P-model environment with penalty probabilities $\{c_1, c_2, \ldots, c_r\}$ for the actions, the average penalty is defined as follows:

$$\begin{aligned} M(n) &= E[x(n) = 1|P(n)] \\ &= \sum_{i=1}^{r} Pr[x(n) = 1|y(n) = y_i]Pr[y(n) = y_i] \\ &= \sum_{i=1}^{r} c_i p_i(n) \end{aligned}$$

where E and Pr denote expectation and probability, respectively, and $p_i(n)$ denotes the probability of taking action i at instant n.
A *pure-chance* automaton is one that chooses each action randomly with equal probability. For such an automaton, the average penalty is a constant $M_0 = \frac{1}{r}\sum_{i=1}^{r} c_i$. Obviously, it is desirable for a learning automaton to perform better than the pure-chance automaton, at least asymptotically as $n \to \infty$. Such an automaton is said to be *expedient* and satisfies the following condition:

$$\lim_{n\to\infty} E[M(n)] < M_0$$

.
An automaton is optimal if $\lim_{n\to\infty} E[M(n)] = c_l$, where $c_l = \min_i\{c_i\}$.

Fixed structure automata

The state transition function F and the output function G of a learning automaton can be stochastic. When all probabilities contained in F and G are constants and independent of time and the inputs, the automaton is referred to as a fixed structure automaton.

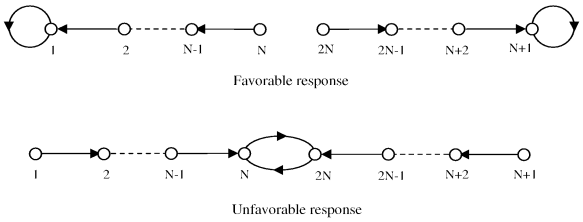


Figure 1: State transition graph for the $L_{2N,2}$ automaton.

$L_{2,2}$: the two-state two-action automaton
In the simplest case, an automaton has two states ϕ_1 and ϕ_2, and two actions α_1 and α_2 with penalty probability c_1 and c_2 respectively. Whenever in state $\phi_i, i = 1, 2$, the automaton takes action c_i with certainty. The so-called $L_{2,2}$ stays in its current state when it receives a favorable response, and change to the other state when an unfavorable response is received.
It is shown in Narendra and Thathachar (1989) that for such an automaton we have

$$\lim_{n\to\infty} M(n) = \frac{2c_1c_2}{c_1 + c_2}.$$

If $c_1 \neq c_2$, we have $\frac{2c_1c_2}{c_1+c_2} < \frac{1}{2}(c_1 + c_2)$, therefore the automaton is expedient.

$L_{2n,2}$: the two-action automaton with memory
An extension to the $L_{2,2}$ automaton is the $L_{2N,2}$ automaton, which has $2N$ states, $\phi_1, \phi_2, \ldots, \phi_{2N}$, and two actions y_1 and y_2 with penalty probability c_1 and c_2. The states can be seen as divided into two groups, each group corresponding to one action. The states $\phi_1, \ldots, \phi_N$ correspond to action y_1 and $\phi_{N+1}, \ldots, \phi_{2N}$ correspond to action y_2.
In state $\phi_i (1 \leq i \leq N)$, the automaton takes action y_1 and based on the response, the state changes as follows:

$$\left.\begin{array}{ll} \phi_i \to \phi_{i+1}, & 1 \leq i \leq N-1 \\ \phi_N \to \phi_{2N} & \end{array}\right\} \text{unfavorable response,}$$

$$\left.\begin{array}{ll} \phi_i \to \phi_{i-1} & (i = 2, \ldots, N) \\ \phi_1 \to \phi_1 & \end{array}\right\} \text{favorable response.}$$

State transitions are defined similarly for states ϕ_{N+1} through ϕ_{2N}. Figure 1 shows the state transition graph for the $L_{2n,2}$ automaton.
We saw in the case of the $L_{2,2}$ automaton, that the action changes whenever one unfavorable response is received. With the $L_{2N,2}$ automaton, it can take up to N consecutive unfavorable responses before a different action is tried, have been received in the current state,

will cause the current action to be taken more times. In this way, the past experience is incorporated into the decision rule of the automaton. The number N is known as the *memory depth* of the automaton.
In Narendra and Thathachar (1989), it is shown that if $min\{c_1, c_2\} \leq 1/2$, then the $L_{2N,2}$ automaton is optimal as $N \to \infty$.

Automaton with multiple actions
The $L_{2N,2}$ automaton can be easily generalized to the K-action ($K > 2$) case. Instead of two groups, the actions are divided into K groups, each corresponding to one distinct action. Upon a favorable response, the state ϕ_i stays unchanged if $i = 1, N+1, 2N+1, \ldots, (K-1)N+1$, or changes to ϕ_{i-1} otherwise. Upon an unfavorable response, the state transition is $\phi_i \to \phi_{i+1}$, if $i \neq N, 2N, \ldots, KN$; otherwise, the transition can be defined in any reasonable manner, for example, $\phi_N \to \phi_{2N}, \ldots, \phi_{(K-1)N} \to \phi_{KN}, \phi_{KN} \to \phi_N$.

Variable structure automata

In a fixed-structure automaton, the state transition or action probabilities are fixed for both favorable and unfavorable responses. For example, in the $L_{2N,2}$ automaton, upon a favorable response, the state stays unchanged if the current state is ϕ_1 or ϕ_{N+1}, or it changes from ϕ_i to ϕ_{i-1} with certainty.
Of greater interest are the more general variable structure automata which allow the state transition or action probabilities to be updated in a way that reflects the responses received. This, in fact, is how learning is achieved since future action is affected by past experience through the modification of the probabilities.
In the following, as previously, we shall assume that each state corresponds to one action. Thus the state probability is the same as action probability. And the dynamics of the learning automaton can be described by a sequence of action probability vectors.

General reinforcement scheme
What differentiates variable structure automata from fixed structure ones is the so-called reinforcement scheme which is an algorithm that determines how the action probabilities are updated as a result of a response from the environment.
Let $P(n) = \{p_1(n), p_2(n), \ldots, p_r(n)\}$ denote the action probabilities of the r actions at instant n of a variable structure automaton. If the action chosen at that instant is $y(n) = y_i$, then a general scheme for updating the action probability is as follows:
If the response $x(n) = 0$ (favorable),

$$\begin{aligned} p_j(n+1) &= p_j(n) - d_j(P(n)), \quad \forall j \neq i, \\ p_i(n+1) &= p_i(n) + \sum_{j \neq i} d_j(P(n)). \end{aligned} \tag{1}$$

If $x(n) = 1$ (unfavorable),

$$\begin{aligned} p_j(n+1) &= p_j(n) + e_j(P(n)), \quad \forall j \neq i, \\ p_i(n+1) &= p_i(n) - \sum_{j \neq i} e_j(P(n)). \end{aligned} \tag{2}$$

In the above, the functions d_j and e_j are continuous nonnegative functions that ensure $p_i(n+1)(1 \leq i \leq r)$ is in the open interval $(0, 1)$. The idea behind the scheme is very simple: if an action results in favorable response, the probability for selecting that action in the future is increased, and the probabilities for other actions are decreased; if the action results in unfavorable response, its probability is decreased, and the probabilities for other actions are increased correspondingly.

The linear reward-penalty (L_{R-P}) scheme
In the linear reward-penalty reinforcement scheme, the updating functions d_i and e_i in equations (1) and (2) are defined as the following linear functions:

$$\begin{aligned} d_j(P(n)) &= a p_j(n), \\ e_j(P(n)) &= \frac{b}{r-1} - b p_j(n)) \end{aligned} \tag{3}$$

where a and b are small positive constants.
It is shown in Narendra and Thathachar (1989) that

$$\lim_{n \to \infty} E[p_i(n)] = \frac{1/c_i}{\sum_{j=1}^{r} 1/c_j} \tag{4}$$

This means the expected probabilities of selecting the actions are in reverse order of the penalty probabilities of the actions.
From (4) we have

$$\lim_{n \to \infty} E[M(n)] = \frac{r}{\sum_{j=1}^{r} c_j} < \frac{1}{r} \sum_{j=1}^{r} c_j = M_0$$

So the automaton using the L_{R-P} scheme is expedient.

The linear reward-inaction (L_{R-I}) scheme
The L_{R-I} scheme is just the L_{R-P} scheme with the number b set to 0. Therefore, the action probabilities are not changed when an unfavorable response is received.
As shown in Narendra and Thathachar (1989), an automaton using the L_{R-I} scheme has the attractive characteristic of being ϵ-optimal, meaning the probability of selecting the action with the lowest penalty probability with certain as $n \to \infty$, i.e., $Pr\{\lim_{n \to \infty} p_l(n) = 1\}$, can be made arbitrarily close to 1.

OPTIMIZING TIME WARP WITH LEARNING AUTOMATA

Adaptive Time Warp is essentially a problem of optimal control of a stochastic process. The environment in the

learning automata system is usually stochastic and finding the optimal action in such an environment is also an optimal control problem. This is basically the motivation for us to apply the learning automata techniques to adaptive Time Warp.
It should be clear from the above discussion, that previous adaptive Time Warp protocols that are based on analytical models essentially assume the adopted models are optimal, whereas with learning automata, we do not assume any initial knowledge of the system and we try to find out the optimal actions through experience. This is the key difference of the learning automata approach from those previous methods.

The algorithms

To formulate optimizing Time Warp as a learning automata-based control problem, we must answer the following questions, among others.

- How many automata do we use?
- What is the control parameter?
- How can the control parameter be mapped into the automaton's actions?
- How do we define the responses from the environment?

Single-automaton vs. multi-automata
Depending on the choice of control parameter, optimizing Time Warp can be formulated as problem involving a single learning automaton or multiple automata. If all of the nodes (a node is a group of one or more LPs running on a single CPU) always share the parameter, then there should only be one automaton. On the other hand, if it is reasonable or even desirable for the nodes to have different values of the parameter at the same time, then multiple automata can be used.
Obviously, the multi-automata setting is much more complex than that of the single-automaton, because each automaton's learning process is affected by others, and as in other learning paradigms that involve multiple learning agents, such as multi-agent reinforcement learning, usually some sort of coordination among the agents is required for the learning to be effective Panait and Luke (2005).
So far in our research we have only implemented the simple case in which there is only one learning automaton. We leave other more complex cases for our future work.

Control parameter
The control parameter we choose for our optimizing Time Warp is a bounded time window, similar to that in Palaniswamy and Wilsey (1993). The window size, W, together with the last GVT defines a limit for event execution. No events scheduled beyond $GVT + W$ are allowed to be executed. If the next scheduled event of an LP is beyond the window, then the LP must block. When blocked, an LP can still receive messages, but it cannot send any messages except for messages involved in GVT computation. When the GVT is updated, the window is moved forward, allowing blocked LPs to unblock if their next scheduled event falls inside of the updated window.
The purpose of the window is to control the optimism of the LPs. It prevents an LP from going too far ahead in virtual time, thereby reducing the risk of long rollbacks.
After deciding what parameter to control, we need to map the parameter to the automaton's actions. A straightforward design is for each action to represent a certain size of the window. We define an incremental unit, U, for the window, such that the window size is always a multiple of U. Then, each action a, $1 \leq a \leq r$, corresponds to adopting a window size of $a * U$.
The window size is updated right after each GVT computation. When the new GVT is obtained, all nodes send their state measurements to a single pre-designated node, which also hosts the learning automaton. Based on the combined data, a new window size is determined by the automaton and broadcast to all nodes.

Response from environment
The response from the environment plays a very important role in the behavior of the learning automaton. As in reinforcement learning Sutton and Barto (1998), the response should directly reflect the goal of the system. For optimizing Time Warp, since the ultimate goal is to reduce the elapsed wall-clock time of the simulation, the reward should be directly related to the speed of the simulation.
Let GVT_n denote the n-th GVT, t_n denote the wall-clock time when GVT_n is computed, and EC_n denote the number of events committed at GVT_n, We define the *event commit rate* (ECR) for the n-th GVT interval (the interval from GVT_{n-1} to GVT_n) as

$$ECR_n = EC_n/(t_n - t_{n-1}).$$

We use ECR_n as the measurement of the simulation speed during the n-th GVT interval.
To define response, we define a reference event commit rate, ECR_{ref}, which is the average event committing rate of a sample period in the very beginning of the simulation with the window size set to its maximal value:

$$ECR_{ref} = (\sum_{i=1}^{D} EC_i)/(t_D - t_0),$$

where D is a small number(a value of 20 has shown to be good enough). In words, ECR_{ref} is the average number of committed events in the D GVT intervals at the beginning of the simulation.

The response for the n-th GVT period, $x(n)$, is: $x(n) = 0$, if $ECR_n \geq ECR_{ref}$; $x(n) = 1$, otherwise.
Therefore, a favorable response is given if the simulation is progressing faster than the reference rate in the most recent GVT period, otherwise, an unfavorable response is given. This way, the speed of the simulation is directly reflected in the reward signal. Of course, digitalizing the relative event commit rate into just two values may be crude. In fact, there are so-called Q- and S-models of environments where the response can take values from a pre-defined range. We leave the study of these models for future work.

Fixed structure automata
With the action and response defined as above, we can readily apply the multi-action fixed structure automata to optimizing Time Warp.
As described above, our automaton has r actions corresponding to the r different window sizes, $U, 2U, \ldots, rU$. Using a memory depth of N, we have rN states for the automaton and start with a random window size $kN(1 \leq k \leq r)$. When a favorable response is received, we have a state transition $\phi_i \rightarrow \phi_{i-1}$, unless i is one of $1, N+1, 2N+1, \ldots, (r-1)N+1$, in which case the state is unchanged. When an unfavorable response is received, the state goes from ϕ_i to ϕ_{i+1}, unless $i = kN(2 \leq k \leq r)$, in which case the state changes to $(k-1)N$, and N will change to rN. This means, whenever it is necessary to try a different window size, we always try the next smaller size, unless the current one is the smallest, in which case we change to the biggest.

Variable structure automata
Using the same definitions for action and response, we can incorporate a variable structure automaton into Time Warp to control parameters like the window size. Unlike the fixed structure automata, variable structure automata achieve learning through the updating of the action probabilities. Initially we assign equal probability to each action and we start with a random action. Using one of the reinforcement schemes discussed in Section , the probabilities of the actions are updated as the simulation proceeds.

EXPERIMENTAL RESULTS

In this section we present test results of a distributed VLSI simulator on two benchmark circuits. The simulator is a Time Warp simulator with no optimizations. Aggressive cancellation is used for rollbacks.
For the bounded time window, we use the clock period as the incremental unit. The automaton's action set is $X = \{1, 2, 3, \ldots, 8\}$, corresponding to setting the window size to x times the clock period.
We have so far implemented the fixed structure learning automata. The test circuits used were the two largest sequential circuits from the ISCAS-89 suite, with about 23,000 gates apiece. In the simulator each logical gate is represented by an LP.

Table 1: Running time in seconds.

Circuits	s38417		s38584	
TW	2606.69		2987.07	
Fixed(N=3)	2282.18	12.5%	2530.64	15.3%
Fixed(N=4)	2270.50	12.9%	2484.65	16.9%

Table 2: Action penalties.

Actions	s38417	s38584
1	0.1144	0.1640
2	0.1185	0.1565
3	0.1716	0.2229
4	0.1990	0.2996
5	0.2732	0.4921
6	0.3639	0.7372
7	0.4598	0.8888
8	0.5970	0.9057

We conducted all of the experiments on a network with 4 AMD Athlon 64 computers running the Linux operating system. The computers were connected by a fast Ethernet switch. MPICH (http://www-unix.mcs.anl.gov/mpi/mpich/) was the underlying message system. Each circuit was given enough random test vectors for them to run for about 40 minutes.

Table 1 shows the simulation time of 3 different automata: fixed structure automata with $N = 3$ and $N = 4$, and variable structure automaton. Also shown are the speedup data compared to the simulation time of pure Time Warp. Each of the simulation times was the average of 10 simulation runs. We can see that using learning automata, we have reduced the simulation time by a percentage of 12.5 to 16.9.

Table 2 shows the action penalty for all 8 actions. For both circuits, the first 2 actions have almost the same penalties.

In Figure 2 we show the evolution of the combined action probability of action 1 and 2, and the penalty probability for the circuit s38417, when a fixed structure automaton was used, with $N = 3$ and $N = 4$ respectively. Figure 3 shows the corresponding charts for s38584.

In both cases, we can see the combined action probability for actions 1 and 2 was higher when $N = 4$ than when $N = 3$. As a result, the penalty probability when $N = 4$ is lower than when $N = 3$. Furthermore the penalty probabilities are very close to the lower bound as $n \rightarrow \infty$: for s38417, they were $0.147(N = 3)$ and $0.138(N = 4)$ with the lower bound 0.114, and for s38584, they were $0.206(N = 3)$ and $0.181(N = 4)$ with the lower bound 0.156.

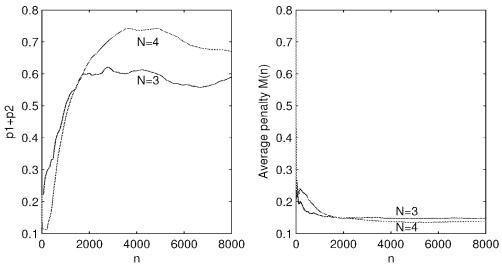


Figure 2: Action probability and average penalty.

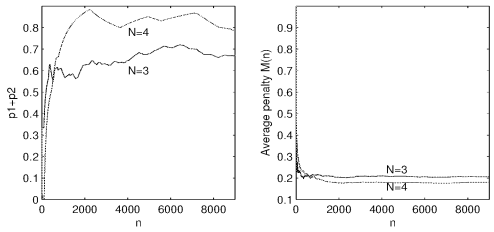


Figure 3: Action probability and average penalty.

CONCLUSIONS

Control techniques for adaptive Time Warp are often based on an analytic model of the system and express the control parameter as a function of some system parameters, making their effectiveness highly dependent on the quality of the model. The underlying assumption is that this pre-defined function provides optimal mapping from system state to the value of control parameter.

In this paper we have presented an alternative approach based on learning automata. Compared with other techniques, our method requires no model of the system at all. The learning automaton has no knowledge of the system or of the control parameter it is supposed to tune. With the only help of the response from the environment, it learns to identify the actions that have the maximum probability of obtaining positive response from the environment. One fact that makes this an attractive method is that the performance of the learning automata is based on solid mathematic foundation. Making use of Bounded Time Warp, we established that the automata could find the optimal window for simulations of two large logic circuits from the ISCAS89 benchmark suite.

For our future work, we intend to implement the variable structure automata, and apply the learning automata techniques to other optimization problems for Time Warp. We also intend to study the more challenging problems such as the Q- and S-model environments and even non-stationary environments. Also of great interest is the multi-automata setting.

REFERENCES

Das S., 2000. *Adaptive protocols for parallel discrete event simulation.* vol. 51, 385–394.

Ferscha A., 1995. *Probabilistic Adaptive Direct Optimism Control in Time Warp.* 120–129.

Fujimoto R., 1997. *Paralle discrete event simulation.* vol. 33, 30–53.

Lin Y.; Preiss B.; Loucks W.; and Lazowska E., 1993. *Selecting the checkpoint interval in time warp simulation.* 3–10.

Lubachevsky B.; A.Shwartz; and A.Weiss, 1991. *An analysis of rollback-based simulation.* vol. 1, 154–193.

Narendra K. and Thathachar M., 1989. *Learning automata: an introduction.* Prentice Hall.

Palaniswamy A. and Wilsey P., 1993. *Adaptive bounded time windows in an optimistically synchronized simulator.* 114–118.

Panait L. and Luke S., 2005. *Cooperative multi-agent learning: the state of the art.* vol. 11, 387–434.

Panesar K. and Fujimoto R., 1997. *Adaptive flow control in time warp.* 108–115.

Reynolds P., 1988. *A spectrum of options for paralle simulation.* 325–332.

Sokol L.; Briscoe D.; and Wieland A., 1988. *MTW: a strategy fo scheduling discrete simulation events for concurrent execution.* vol. 19, 34–42.

Sutton R. and Barto A.G., 1998. *Reinforcement learning: an introduction.* The MIT Press.

ELECTRICAL SIMULATION

Projection Methods for Integral Equations in the Modelling in the Electrical Engineering

Lechosław Hącia and Adam Marlewski
Institute of Mathematics, Faculty of Electrical Engineering
Poznań University of Technology
ul. Piotrowo 3a,
60-965 Poznań,
Poland
E-mails: lhacia@math.put.poznan. pl, amarlew@math.put.poznan.pl

KEYWORDS
Current density, Fredholm integral equation, heat conduction problems, radiosity problems, Volterra-Fredholm integral equation.

ABSTRACT

Such problems in the electrical engineering as the current density, the radiative heat transfer and the heat conduction can be mathematically described by integral equations, namely by Fredholm integral equations, integro-differential equations and Volterra-Fredholm integral equations, respectively. In this paper we discuss some methods to solve these equations and we propose a projection method of Galerkin type. In this method we replace the initial equation by a system of Volterra integral equations and, next, we solve it numerically. Discussed techniques are accompanied by exemplary computational results.

INTRODUCTION

In the electrical engineering the radiative heat transfer, the current density problems and Fourier problems can be mathematically covered by Fredholm integral equations, integro-differential equations and Volterra-Fredholm integral equations, respectively. The exact solutions being impossible to be determined, their numerical approximations can be produced using various techniques, in which original/initial equations are replaced by systems of linear algebraic equations. Among these techniques we find the projection methods, which are based on the linear combinations of apriori chosen functions. We mention some such schemes (Galerkin method locates within this area) and we propose an other projection method as well as its numerical treatment (to reduce the number of indexes from 4 to 2).

INTEGRAL MODELLING IN THE CURRENT DENSITY THEORY

In [7] there are given Fredholm integral equations of the second kind covering a time-constant current density in the conductor with rectangular cross-section, and there is discussed the discretization method to solve these equations. In this approach the system is approximated by a system of linear algebraic equations, and this approximation can be get by various methods, e.g. iterative methods, Gauss method, Croute method. In [4] and [5] there is considered the current density problem varying in time. It is described by a system of integro-differential equations. By appropriate discretization w.r.t. both time and space variables that system of equation is replaced by a system of linear algebraic equations. In this paper we generalize these techniques. Moreover, we propose methods to produce approximative solutions to Fredholm integral equations [2] and integro-differential equations, and we achieve it by the discretization in time in aim to have Fredholm integral equations.

As in [4], [5], [7]-[9], we deal with the mathematical model of m nonmagnetic conductors which are situated parallel to z-axis of the rectangular coordinate system $Oxyz$ and have cross-sections S_k (k = 1, 2, ..., m). Through these conductors there flow currents $i_1(t)$, $i_2(t)$, .., $i_m(t)$. The magnetic vector potential $\mathbf{A}$ generated by them is parallel to the z-axis and independent of z. The conductors are considered to be infinitely long. Under these assumptions the problem is two dimensional and it is governed by

$$\mathbf{J} = \mathbf{1}_z J(x,y,t),$$
$$\mathbf{A} = \mathbf{1}_z A(x,y,t),$$

where $\mathbf{J}$ stands the current density vector and, $\mathbf{1}_z$ is the unit vector along the z-axis.

If, as in [0], within each conductor there is considered one reference point (x_{ok},y_{ok}), the state of the scalar density J is governed by system of integro-differential equations:

$$J(x,y,t) - J(x_{ok},y_{ok},t) + \frac{\gamma\mu_o}{2\pi}\cdot\frac{\partial}{\partial t}\int_S J(x',y',t)\cdot\ln\frac{r_{ok}}{r}\,dx'dy' = 0, \qquad (1)$$

where γ and μ_0 are physical constants, the conductivity and the magnetic inductive capacity, resp., and k = 1, 2, ..., m.

The total current flowing through the k-th conductor is

$$i_k(t) = \int_{S_k} J(x',y',t)\,dx'dy', \qquad (2)$$

k = 1, 2, ..., m.

System (1)-(2) of equations completed, we discretize them in time and space similarly as in [5], i.e. we use the forward divided difference with respect to time t at $t = t_n$,

$$\frac{\partial}{\partial t}J(x',y',t_n) \approx \frac{\Delta J(x',y',t_n)}{\Delta t_n} == \frac{J(x',y',t_n) - J(x',y',t_{n-t})}{t_n - t_{n-1}}$$

with fixed x', y', and we assume that

$$J(x,y,t_o)=\frac{i(t_o)}{S}=\text{const}.$$

This way we replace the system (1) - (2) by the system

$$\begin{cases} J(x,y,t_n)-J(x_{ok},y_{ok},t_n)+ \\ \qquad \dfrac{\gamma\cdot\mu_o}{2\pi}\cdot\dfrac{1}{\Delta t_n}\cdot\displaystyle\int_S\left(J(x',y',t_n)-J(x',y',t_{n-1})\right)\cdot\ln\frac{r_{ok}}{r}\,dx'dy'=0, \\ \qquad\qquad \displaystyle\int_{S_k}J(x',y',t_n)\,dx'\,dy'=i_k(t_n), \end{cases}$$

which is composed of Fredholm integral equations. Approximating w.r.t. the space variables x, y, see [2] and [3], we turn the above system into a system of linear algebraic equations. Its numerical solution gives the values of the current density J at the moment t_n and at apriori chosen spatail points if there are given values of J at the previous while t_{n-1}.

Example 1. Figure 1 visualizates the numerical results for a distribution of the current density J at the time t = 0.1 s if in the conductor of the size 1 cm × 1 cm of the conductivity γ = 56 MS/m there flows the current $i(t) = 10\cdot t$. The regular computational mesh was used with 20×20 = 400 cells S_k.

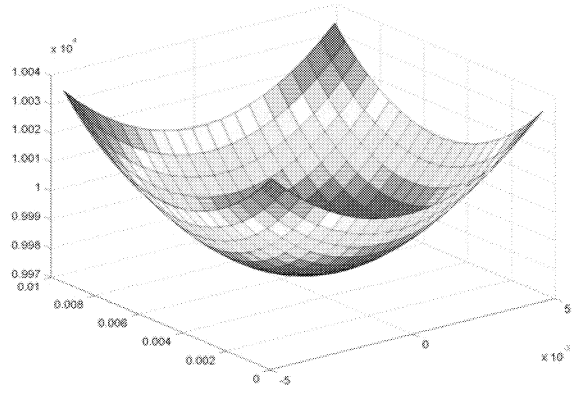


Figure 1:The distribution of the current density at t = 0.1 s

Remark 1. Current density problems stable in time are mathematically modelled by systems of Fredholm integral equations [7, 8], and that varying in time by integro-differential equations [5], both these groups of equations by an appropriate discretization are approximated by systems of algebraic equations.

FREDHOLM INTEGRAL EQUATIONS IN THE HEAT TRANSFER PROBLEMS

Ray tracing methods are based on the rendering equation, see e.g. [1],

$$I(x,x')=g(x,x')\cdot\left[e(x,x')+\int_V\rho(x,x',x'')\cdot I(x',x'')\,dx''\right] \quad (3)$$

where $I(x, x')$ is the sum of emitted and reflected radiant intensities at point x' towards point x; $g(x, x')$ is a geometrical factor; $e(x, x')$ is the emission of radiation from point x' towards point x; $\rho(x, x', x'')$ is the bi-directional reflectance distribution function, V is the area where the phenomena occurs (e.g. within an area in the Euclidean space R^3).

The equation describing heat balance of point x^0 in Siegel and Howell [10] has a form

$$p_{eff}(x^0,T,\theta,\phi)=p_e(x^0,T,\theta,\phi)+ \int_\Omega\rho(x^0,\theta,\phi,\theta_{in},\phi_{in})\cdot p_{in}(x^2,T_j,\theta_{in},\phi_{in})\cdot\cos\theta\,d\omega \quad (4)$$

where, in the radiative heat transfer terminology, p_{eff} stands for the surface density of the effective radiant intensity (radiance) of point x^0 in the direction of x^1, defined by angles (θ, ϕ); T represents the temperature and the index *in* concerns the incident radiation.

Remark 2. Problems in the hear transfer are described by systems of integral equations of the Fredholm type. These equations, by an appropriate discretization, are replaced by systems of linear algebraic equations, as it is presented in [1], [2] and [9]. Notice that Fredholm integral equations are a particular case of integral equations which arise in the Fourier theory, see [3] and [4].

Equations (3) or (4) are Fredholm integral equations of the second kind of the form

$$y(\mathbf{r})=f(\mathbf{r})+\int_S k(\mathbf{r},\mathbf{r}')\cdot y(\mathbf{r}')\,dS,$$

where $k(\mathbf{r}, \mathbf{r}')$ is called a kernel of that equation, $f(\mathbf{r})$ is a given function and $y(\mathbf{r})$ is the solution (e.g. the radiance L, the radiance monochromatic density l or the heat flux density p) we want to determine.

Numerical methods for Fredholm integral equations and their systems are presented in [4]-[7]. In particular, in [6] there is discussed an iterative-collation method to get the solution. In the next part of this paper we deal with integral equations considered in the space-time and we propose a new projection method.

INTEGRAL EQUATIONS IN THE MODELLING OF FOURIER PROBLEMS

We consider the integral equation of the mixed type

$$u(x,t)=f(x,t)+\int_0^t\int_M k(x,t,y,s)\cdot u(y,s)\,dy\,ds \quad (5)$$

or shortly

$$u=f+Ku \quad (6)$$

which compasses Volterra integral equation (w.r.t. t) and Fredholm integral equation (w.r.t. the spatial variable x). The equation at hand play a very important role in some problems in the electrical engineering, it arises in the heat conduction theory. Some initial-boundary problems for a number differential partial equations in physics are also reducible to this equation.

Let's consider this equation in the space-time, where f is a given function in the domain $D = M \times [0, T]$, M is a compact subset of m-dimensional Euclidean space, u is an unknown function defined on D. The kernel k is defined in the domain

$$\Omega = \{(x,t,y,s) : x, y \in M, 0 \le s \le t \le T\}.$$

Let us illustrate the Galerkin method for the equation of the form

$$u(x,t) = f(x,t) + \int_0^t \int_{-1}^{1} k(x,t,y,s) \cdot u(y,s)\, dy\, ds, \quad t \in [0,1] \quad (7)$$

when the orthonormalized Legendre polynomials are used.

Example 2. The Galerkin solution of equation (7) with

$$k(x,y,t,s) = x^2 t^2 e^s$$

and

$$f(x,t) = x^2 \cdot \left(e^{-t} - \frac{2}{3} t^3 \right)$$

and 6 basic functions gives results carrying relative errors listed in below table. x

Table 1. Relative errors in Example 2 for n = 6

$x \setminus t$	0.0	0.2		0.6	0.8	1.0
-1.0	.9619e-6	.2089e-6		.1273e-6	.3487e-6	.2317e-5
-0.6	.9625e-6	.2093e-6		.1270e-6	.3480e-6	.2315e-5
-0.2	.9685e-6	.2146e-6		.1266e-6	.3432e-6	.2300e-5
0.2	.9685e-6	.2146e-6		.1266e-6	.3432e-6	.2300e-5
0.6	.9625e-6	.2093e-6		.1270e-6	.3480e-6	.2315e-5
1.0	.9619e-6	.2089e-6		.1273e-6	.3487e-6	.2317e-5

Example 3. We have the equation

$$u(x,t) = x \sin t + \frac{1}{3}\left(x^2 t^2 \cdot e^t \cos t - e^t \sin t - 1\right) + \int_0^t \int_{-1}^{1} y x^2 t^2 e^s u(y,s)\, dy\, ds \quad (8)$$

The absolute errors of the Galerkin solution are shown in Figure 2 for two values n (the number of basic functions), n = 3 and n = 5.

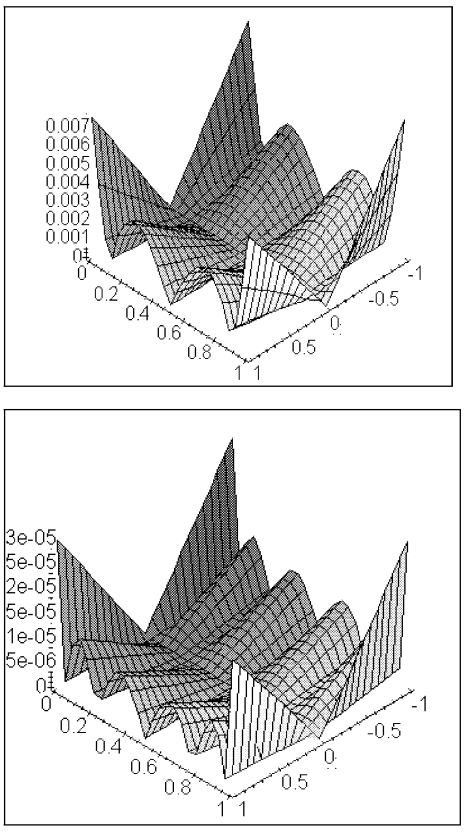


Figure 2. Relative errors in Example 3

PROJECTION METHOD WITH A SYSTEM OF VOLTERRA INTEGRAL EQUATIONS

Taking an arbitrary basis composed of functions $\chi_{ik} = \chi_{i,k}(x, t)$, we replace the equation (5) by the equation

$$u_n(x,t) = f_n(x,t) + \int_0^t \int_M k_n(x,t;y,s) u_n(y,s) dy ds, \quad (9)$$

where

$$f_n(x,t) = \sum_{i,k=1}^{n} f_{ik} \chi_{ik}(x,t), \quad (10)$$

$$k_n(x,t,y,s) = \sum_{i,k=1}^{n} \sum_{j,l=1}^{n} k_{ik,jl} \chi_{ik}(x,t) \chi_{jl}(y,s), \quad (11)$$

$$k_{i,k,j,l} = (k, \chi_{lk,jl}) = \int_0^T \int_M \left[\int_0^t \int_M k(x,t;y,s) \cdot \chi_{jl}(y,s)\, dy\, ds \right] \cdot \chi_{ik}(x,t)\, dx\, dt. \quad (12)$$

The structure simplifies if the basis $(\chi_{ik})_{i,k=1,2,\ldots,n}$ is orthonormal in the space $L^2(D)$. Then (9) writes

$$u_n(x,t) = \sum_{i,k=1}^{n} \left[f_{i,k} + \sum_{j,l=1}^{n} k_{ik,jl} \int_0^t \int_M \chi_{jl}(y,s) u_n(y,s) dy ds \right] \chi_{ik}(x,t)$$

and we have the approximate solution in the form

$$u_n(x,t) = \sum_{i,k=1}^{n} c_{ik}(t) \chi_{ik}(x,t), \quad (13)$$

where

$$c_{i,k}(t) = f_{i,k} + \sum_{j,l=1}^{n} k_{i,k,j,l} \cdot \int_0^t \int_M \chi_{j,l}(y,s) u_n(y,s)\, dy ds$$

Taking into account the representation (13) we have

$$c_{ik}(t) = f_{ik} + \sum_{j,l=1}^{n} k_{ik,jl} \cdot \int_0^t \int_M \chi_{jl}(y,s) \left[\sum_{p,q=1}^{n} c_{pq}(s) \chi_{pq}(y,s) \right] dy ds$$

and, consequently,

$$c_{ik}(t) = f_{ik} + \sum_{j,l=1}^{n} k_{ik,jl} \sum_{p,q=1}^{n} \int_0^t \int_M c_{pq}(s)\cdot\phi_j(y)\cdot\psi_l(s)\cdot\phi_p(y)\cdot\psi_q(s)\,dy\,ds,$$

if basic functions are products of the form

$$\chi_{ik}(x,t) = \varphi_i(x)\cdot\psi_k(t). \qquad (14)$$

If the system $(\varphi_p)_{p=1,2,\ldots,n}$ is orthonormal in $L^2(M)$, i.e.

$$(\varphi_j, \varphi_p) = \int_M \phi_j(y)\cdot\phi_p(y)\,dy = \delta_{j,p},$$

we obtain the system of Volterra integral equations

$$c_{ik}(t) = f_{ik} + \sum_{l,p,q=1}^{n} k_{ik,pl} \cdot \int_0^t c_{pq}(s)\cdot\psi_l(s)\cdot\psi_q(s)\,ds. \qquad (15)$$

As we see, the projection method reduces the initial equation (5) to the system (15) and this system has exactly one solution in the class $L^2([0, T])$.
Equations (9) can be rewritten in the operator form

$$u_n = f_n + K_n u_n, \qquad (16)$$

where K_n is Volterra-Fredholm integral operator of form (6) determined by kernel k_n (11).

Now we can prove the following
Theorem
Let $\{\varphi_i\}$ be orthonormal complete system in $L^2(M)$ and $\{\psi_i\}$ be an orthonormal complete system in $L^2[0,T]$. If $f \in L^2(D)$ and $k \in L^2(\Omega)$, then the system (14) is uniquely solvable and sequence $\{u_n\}$ defined by formula (13) converges, as $n \to \infty$, in the space $L^2(D)$ to the unique solution of equation (5). Moreover, the estimate error

$$\| u_n - u \|_{L^2(D)} \le \frac{c}{1 - c\cdot d_n}\left[\| f - f_n \|_{L^2(D)} + \| u \|_{L^2(D)} \cdot d_n\right]$$

holds with

$$c = \left\| (I-K)^{-1} \right\|_{L^2(\Omega)}$$

and $\quad d_n = \| k_n - k \|_{L^2(\Omega)}$.

Sketch of the proof. Let us notice

$$u_n - u = f_n - f + (K_n - K)u_n + K(u_n - u).$$

Hence

$$\|u_n - u\|_{L^2(D)} \le \left\| (I-K)^{-1} \right\| \cdot \left\{ \|f_n - f\|_{L^2(D)} + \|k_n - k\|_{L^2(\Omega)} \|u_n\|_{L^2(D)} \right\}$$

To obtain an error estimate let us notice that

$$\|u_n\|_{L^2(D)} \le \|u_n - u\|_{L^2(D)} + \|u\|_{L^2(D)},$$

where $u = (I-K)^{-1} f$ is an unique solution of equation (5), I stands for the identity operator.
By the above considerations we have

$$\|u_n - u\|_{L^2(D)} \le c \cdot \left\{ \|f_n - f\|_{L^2(D)} + d_n \|u_n\|_{L^2(D)} \right\} \le$$
$$c \cdot \|f_n - f\|_{L^2(D)} + c \cdot d_n \cdot \left\{ \|u_n - u\|_{L^2(D)} + \|u\|_{L^2(D)} \right\}$$

From here we get the estimation.
By completness of systems (φ_p) and (ψ_q) we have

$$\|f_n - f\|_{L^2(D)} \xrightarrow[n\to\infty]{} 0,$$
$$\|k_n - k\|_{L^2(\Omega)} \xrightarrow[n\to\infty]{} 0,$$

and this way the convergence is proved.

To get an idea how the method works let's see following **Example 4**. We consider the integral equation solved in example 3 by Galerkin method leading to a system of algebraic equations. We use the proposed projection method with a system of Volterra integral equations with h – the length of integration steps and n – the number of basis functions.

Table 2: Relative errors in Example 4

$h = 10^{-2}$, $n = 5$:

x	t				
	0,2	**0,4**	**0,6**	**0,8**	**1,0**
-1,0	$0{,}526\cdot10^{-11}$	$0{,}424\cdot10^{-10}$	$0{,}144\cdot10^{-9}$	$0{,}342\cdot10^{-9}$	$0{,}6709\cdot10^{-9}$
-0,5	$0{,}008\cdot10^{-11}$	$0{,}016\cdot10^{-10}$	$0{,}025\cdot10^{-10}$	$0{,}033\cdot10^{-10}$	$0{,}0408\cdot10^{-10}$
0	$0{,}471`\cdot10^{-11}$	$0{,}379\cdot10^{-10}$	$0{,}128\cdot10^{-9}$	$0{,}306\cdot10^{-9}$	$0{,}6001\cdot10^{-9}$

$h = 10^{-3}$, $n = 5$:

x	t				
	0,2	**0,4**	**0,6**	**0,8**	**1,0**
-1,0	$-0{,}157\cdot10^{-35}$	$0{,}157\cdot10^{-34}$	$0{,}471\cdot10^{-44}$	$-0{,}157\cdot10^{-34}$	$-0{,}314\cdot10^{-34}$
0	$-0{,}140\cdot10^{-35}$	$0{,}141\cdot10^{-34}$	$0{,}421\cdot10^{-44}$	$-0{,}141\cdot10^{-34}$	$-0{,}281\cdot10^{-34}$

Results are much better than that produced in Example 3.

NUMERICAL TREATMENT

Introducing the notation

$$P_{ik,pq}(s) = \sum_{l=1}^{n} k_{ik,pl}\ \psi_l(s)\psi_q(s)$$

we can write the system (15) in the form

$$c_{ik}(t) = f_{ik} + \sum_{p,q=1}^{n} \int_0^t P_{ik,pq}(s)\cdot c_{pq}(s)ds, \quad (17)$$
$$i, k = 1, 2, \ldots, n,$$

or

$$d_m(t) = B_m + \sum_{r=1}^{n^2} \int_0^t E_{mr}(s)\cdot d_r(s)ds,$$

where

$$B_m = f_{ik},$$
$$E_{rm}(s) = P_{ik,pq}(s),$$
$$d_m(t) = f_{ik}(t),$$
$$m = (i-1)n + k \qquad \text{for } m = 1,2,...,n^2,$$
$$r = (p-1)n + q \qquad \text{for } r = 1,2,...,n^2.$$

To calculate values $d_m(t_l)$ for $m = 1, 2, ..., n^2$ at apriori chosen points t_l ($l = 1, 2, ..., m$) we apply a quadrature formula with coefficients A_p, $p = 0, 1, ..., l$. This way we produce values

$$d_m(t_l) = B_m + \sum_{r=0}^{n^2} \sum_{p=0}^{l} A_p \cdot E_{mr}(t_p) \cdot d_j(t_p).$$

Completing vectors B and $D(l)$ and the matrix $Q(p)$ as follows

$$B = \left[B_1, B_2, ..., B_{n^2-1}, B_{n^2}\right]^T$$
$$D(l) = \left[d_1(t_l), d_2(t_l), ..., d_{n^2}(t_l)\right]^T.$$
$$Q(p) = \begin{bmatrix} E_{00}(t_p) & E_{01}(t_p) & ... & E_{0n^2}(t_p) \\ E_{10}(t_p) & E_{10}(t_p) & ... & E_{1n^2}(t_p) \\ ... & ... & ... & ... \\ E_{n^20}(t_p) & E_{n^21}(t_p) & ... & E_{n^2n^2}(t_p) \end{bmatrix}$$

we can write the system in the form

$$D(l) = B + \sum_{p=0}^{l} A_p \cdot Q(p) \cdot D(p) =$$
$$= B + \sum_{p=0}^{l-1} A_p \cdot Q(p) \cdot D(p) + A_l \cdot Q(l) \cdot D(l),$$

so

$$(I - A_l \cdot Q(l)) D(l) = B + \sum_{p=0}^{l-1} A_p \cdot Q(p) \cdot D(p),$$

where I is the unit matrix. Therefore, if $I - A_l \cdot Q(l)$ is invertible, there is

$$D(l) = \left[I - A_l \cdot Q(l)\right]^{-1} \left[B + \sum_{p=0}^{l-1} A_p \cdot Q(p) \cdot D(p)\right], \qquad l = 2,...,m,$$

and $D(1) = B$.

Then, by (13) and (14),

$$u_n(x,t) = \sum_{i,k=1}^{n} c_{ik}(t) \cdot \varphi_i(x) \cdot \psi_k(t)$$

is an approximate solution of equation (5) with normalized Legendre polynomials φ_i on the interval [–1, 1], i=1,2,...,n, and ψ_k on the interval [0, 1], k=1,2,...,n.

Remark 3. Differentiating the equations in (16) with respect to the time variable t we obtain the system of differential equations

$$c_{ik}'(t) = \sum_{p,q=1}^{n} P'_{ik,pq}(t) \cdot c_{pq}(t)$$

with initial conditions

$$c_{ik}(0) = f_{ik}, \ i, k = 1, 2, ..., n.$$

This suggests that instead of application of numerical quadrature formulas one can derive the approximative solution using e.g. Runge-Kutta methods.

CONCLUSION

In this paper we restricted to three problems in electrical engineering, namely to the radiative heat transfer, current density problems and Fourier problems (i.e. concerning the heat conduction), in the aspect of their solution if they are covered by systems of Fredholm integral equations, integro-differential equations and Volterra-Fredholm integral equations, respectively. We mentioned various approaches to the question and we focused on the proposal to apply projection methods which results in systems of algebraic equations or systems of integral equations of the Volterra type. Numerical experiments promise a well future to this technique, and up-to-now obtained theoretical estimations let hope that this approach works fine (although the question of the rounding errors still have to be investigated).

REFERENCES

[1] Jaluria, Y. and Torrance K.E. 1986. *Computational heat transfer*, Hemisphere Pub. Co., Washington.

[2] Hącia, L. and Domke K. 2006. "Fredholm integral equations in the radiative heat transfer problems", *Academic Journals – Electrical Engineering* 52, 115-123.

[3] Hącia, L. 2002 "Computational methods for Volterra-Fredholm integral equations", *Computational Methods in Science and Technology*, Vol 8, No 2, 13-26,.

[4] Hącia, L. 2004. "Some integral mathematical models in electrotechnics", *Computer Applications in Electrical Engineering-Monograph* 1, Wyd. PP, Poznań, , 106-123.

[5] Hącia, L. and Patecki A. 2003. "Analysis of the current density in the conductor by integro-differential equations", *VIII Conference on Applications of Computers in Electrical Engineering,* Poznań/Kiekrz, 159-162

[6] Hącia, L. 2007. **"**Iterative-collocation method for integral equations in the heat conduction problems", *Lecture Notes in Computer Science*, 4310, 378-385.

[7] Nawrowski, R. and Pabian, J. 2002. „Obliczanie rozkładu gęstości prądu w przewodach metodą równań całkowych Fredholma", *VII Konferencja Naukowo-Techniczna – Zastosowanie Komputerów w Elektrotechnice,* Poznań/ Kiekrz, 183-186.

[8] Nawrowski, R., Pabian, J. and Tomczewski, A. 2006. "The use of object-oriented software for computation of current density distrubution", *Academic Journals – Electrical Engineering* 52, 131-144.

[9] Szymański, G. 1983. *Zastosowanie metod całkowych w elektrodynamice technicznej. Rozprawy nr 144*, Wyd. PP, Poznań.

[10] Siegel, H. and Howell, J.R. 1972. *Thermal radiation heat Transfer*, Mc-Graw Hill Book Co., New York.

Kahn Based Performance Model Within Co-Design Flow

Kamel Smiri
Mohamed Moalla
LIP2 Laboratory,
Faculty of Sciences of Tunis, Tunisia.
Campus University, 2092 Manar 2,
Tunisia.

Hajer Harbegue
Abderrazek Jemai
LIP2 Laboratory,
Faculty of Sciences of Tunis, Tunisia.
INSAT, B.P. 676, 1080 Tunis
Cedex, Tunisia.

Ahmed Chiheb Ammari
MMA Unit of Research,
INSAT, B.P. 676, 1080
Tunis Cedex, Tunisia.

E-mail: smiri_kamel@yahoo.fr

KEYWORDS
Disydent, Performance Model, Decoder Motion-JPEG, Co-design, MPSoC.

ABSTRACT

Current multimedia applications require high computation capacities. Their implementation on Multiprocessor Systems on Chip (MPSoC) targets to answer the performance constraints. For this study, we have first experiment the design method of the framework Disydent, using an example of multimedia application (MJPEG decoder), in order to find an optimal architecture, verifying functional and non-functional constraints. Second, we aimed to criticize the Disydent approach which does not consider performance. A performance model based on the Kahn process network is proposed to enrich the Disydent flow in term of temporal performance evaluation.

INTRODUCTION

The concept of Co-design uses both hardware and software in the design process. Some authors focus much more on hardware design than software and vice versa. Co-design covers different domains. The final implementation may be different systems such as field-programmable Gate Array (FPGA) board, digital signal processor (DSP) board with dedicated coprocessors or a multiprocessor system on chip (MPSoC). In the literature (Duntrascu and al. 2006) (Bonacui and al. 2006), we distinguish three MPSoC design approaches: (1) platform based design, (2) component based design and (3) system synthesis. Current multimedia applications require heterogeneous MPSoC architectures in order to achieve computation and communication performances. Heterogeneous MPSoC includes different kinds of processing units (DSP, microcontroller, etc) and different communication schemes (fast links, non standard memory organisation and access) (Popovici and al. 2007).

In this paper, we are interested in the platform based design which targets implementation of an application on a configurable prototype of an architecture. It's about a multimedia application characterized by an important computation and high capacity of data transfer (Marculescu and Eles 2000). Many codesign techniques are actually used. We can extract instruction level parallelism (ILP) and generate multiple instructions on single data (MISD) machine (Faraboschi and al. 2000). Other works (Quinton and Van Dongen 1989) use spatial and temporal projections on the loop nest to generate simple instruction on multiple data (SIMD) machines. Another approach based on task level parallelism extracts coarse grain parallelism to generate multiple instructions on multiple data (MIMD) machines (Vercautern and al. 1996).

This document is organized as follows: Section II presents Disydent as a framework for the co-design of embedded systems. In section III, we presented the results of an example of implementation using an MJPEG decoder on "asim0" platform. Section IV describes the optimal architecture which verifies the constraints. Finally, we proposed a performance model based to enrich Disydent of temporal performance evaluation.

FRAMEWORK DISYDENT

Disydent[†] is a framework dedicated to multiprocessor system on chip (MPSoC) platform-based design for shared memory Multiple Instructions Multiple Data (MIMD) architecture (LIP6 2006). The final system is an extension of a configurable architecture prototype. Embedded application implementation must verify functional constraints about the application and non functional constraints about hardware architecture. The integrated application is modelled using a parallel computation model such as YAPI (Brunel and al. 2000), KPN (Kahn 1974), etc. Disydent uses Kahn Process Network (KPN). This computation model describes independent concurrent processes communicating using lossless FIFO channels and uses blocking reading for synchronization (KPN 2006). Application parallel model is implemented using Disydent Process Network (DPN) C-library, which offers primitives and services for POSIX threads management and data exchange.

An extended MPSoC system includes both hardware and an operating system. This system is based on an "asim0" platform, containing n >=1 MIPS processors, a Pi-Bus interconnexion bus, a FIFO communication interface, an interruption controller, direct memory access controllers, and to which we add hardware accelerators (or coprocessors). Asim0 contains. The operating system «Mutek» is a multiprocessor multithread kernel supporting POSIX APIs. It proposes a communication layer and includes a dynamic SMP[‡] scheduler and a static scheduler (Non SMP-CS/DS[§]).

The Disydent design flow is based on four main phases to transform an application from a sequential specification to an optimal implementation model verifying constraints (Augé and al. 2004). It starts by a sequential implementation, consisting in executing a sequential specification of the application on a simulated platform. If constraints are not met, the designer moves up to the next step to implement the parallel specification

[†] Disydent is an open framework developed by the ASIM department of the LIP6 laboratory

[‡] Symmetric Multiprocessor

[§] Non SMP Centralized Scheduler/Distributed Scheduler

based on the KPN model. The KPN specification is implemented on a multiprocessor platform. If the software implementation of this model does not realise constraints, then the designer looks for a suitable Hardware/Software mapping of the application. At this step, groups of tasks to migrate in hard must be identified. The last step is synthesis which consists in synthesis and validation of processes mapped in hardware using UGH tool (Augé and al. 1997).

EXPERIMENTATION: MOTION-JPEG DECODER

We have validated the Disydent design flow using a Motion JPEG decoder. We want to find an implementation of the MJPEG decoder realising 25 frames per second (fps) as a functional constraint and using 50MHz processors as a non-functional constraint. We are only interested in studying temporal performance depending on the three following aspects (1) thread execution time, (2) bus transfer rate and (3) the transfer primitive latency.

The MJPEG decoder reads a stream of JPEG images from an input peripheral (Traffic Generator or TG) and writes pixels on an output peripheral named RAMDAC. The first functional bloc DEMUX dispatches the input stream to the other blocs. The decoding chain begins with the decompression bloc (VLD), then the zigzag rearrangement bloc (ZZ), followed by the inverse quantification bloc (IQ). Finally, the stream passes to the inverse discrete cosine transformation (IDCT) (JPEG).

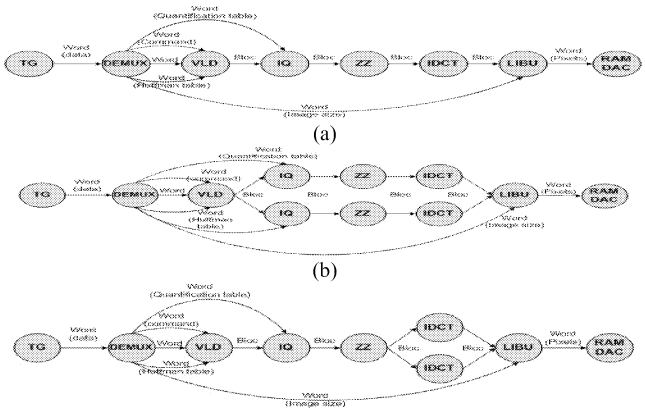


Figure 1: (a) KPN model of Motion JPEG decode, (b) 1st parallelization solution, (c) 2nd parallelization solution

The sequential execution of the decoder on a multiprocessor platform realises 12.2 fps. Then, we need to duplicate the decoding performance by 2.6. The second step of the design flow is the parallel implementation. We have used three parallel models shown in figure1.

Software implementation of the KPN model (Figure 1.a) on the asim0 platform with 1, 2, 3 and 4 MIPS processors, using the SMP kernel, had sped up the decoding process. But the execution time stagnates with the fourth processor at the value of 13.86 fps. We have been able to evaluate the complexity of each task by using profiling tools. The most important task is IDCT with 72% of the execution time, followed by the VLD task with 11.3%.

The model described in figure 1.b is parallelized at bloc level, by duplicating IQ, ZZ and IDCT tasks. Using the asim0 platform with 5 processors and the SMP kernel of the «Mutek» (dynamic scheduler), this model reaches 23.35 fps. This implementation model presents an inconvenient: important communication time between processors. The double IDCT model (Figure 2.c) has accelerated the execution time and reaches 23.3 fps with 5 processors.

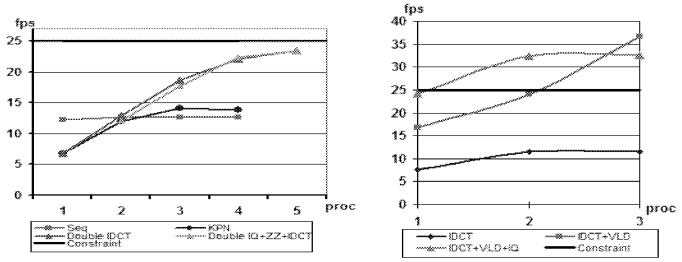


Figure 2: (a) Performance of the software implementation, (b) Performance of the hardware implementation of process groups.

In our work, we took into consideration data and instruction memory cache size. The designer must find a compromise between memory cache size and processors numbers. Because the functional constraint is not yet satisfied, we moved up to a hardware/software mapping. Using tasks complexity profile and using the software implementation results, we have selected 3 groups of processes: (IDCT), (IDCT+VLD) and (IDCT+VLD+IQ). Hardware implementation of group (IDCT) has realized 11.55 fps with 3 processors on the platform. So, we need an execution speedup of 2.5. By mapping group (IDCT+VLD) into hardware, the constraint was almost satisfied using 2 processors on the platform (24.07 fps). In this case, a code optimisation and a suitable configuration of the platform are necessary to reach 25 fps. The (IDCT+VLD+IQ) group had given as execution performance 24.12 fps with only one processor on the platform (Figure 2.b).

OPTIMAL ARCHITECTURE FOR MJPEG DECODER

The optimal architecture released is relative for the constraints fixed in the beginning. The first constraint is the functional constraint which is an execution speed equal to 25 fps. The non-functional constraint is related to the platform which is using MIPS processors with 50 MHz.

Table 1: Implementation Performance of the Decoder MJPEG

proc	SMP-CS				Non SMP-CS		SMP - CS		
	Seq	KPN	ZZ + IQ + IDCT	Double IDCT	Double IDCT	ZZ + IQ + IDCT	IDCT hard	IDCT+VLD hard	IDCT+VLD+IQ hard
1	12,2	6,72	---	6,78	---	---	7,7	17	24,12
2	12,61	12,01	12,16	12,86	---	12,68	11,49	24,07	32,45
3	12,63	14,1	17,65	18,63	30,49	---	11,55	36,73	32,56
4	12,6	13,86	22,35	22,09	25,53	---	---	---	---

Table 1 presents the different results concerning the implementation performance of the Motion JPEG decoder in frames per second. The purpose is to find out an optimal architecture. We had changed platform characteristics: processor number, memory cache size and the scheduler of the "Mutek". Among the process groups mapped into hardware, we had chosen the group (IDCT+VLD), which realized 24.07 fps with 2 MIPS (Figure 3).

The optimal architecture contains a Pi-Bus interconnexion bus, an interruption controller, memory, 2 MIPS processors with 4 Ko data and instruction memory cache. The platform also contains an input peripheral for reading video stream and an output peripheral for displaying the decoder witch contains 4 software tasks: DEMUX, IQ, ZZ and LIBU. Communication between software and hardware components is ensured by a VCI interfacing module (Petrot 2002). Disydent framework is considered as a semi-automatic MPSoC design method. However, Disydent targeted architectures (asim0) have its components plugged on a Pi-Bus communication system. Besides, Disydent design method doesn't consider functional and non functional constraints.

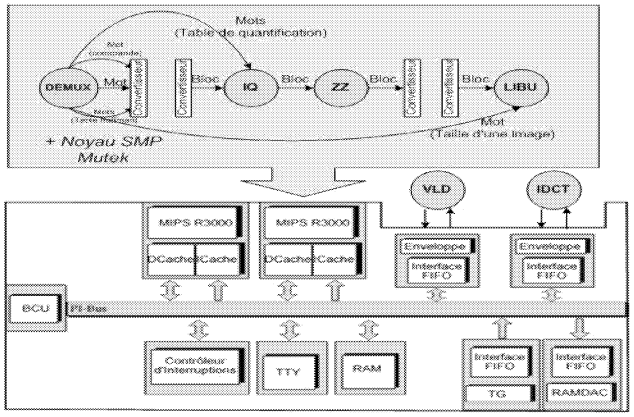


Figure 3: Optimal architecture for Motion JPEG decoder.

PROPOSAL OF A PERFORMANCE MODEL FOR DISYDENT CO-DESIGN

The Disydent approach is not based on adequate abstraction levels like in SystemC (Untimed Fonctional, Timed Fonctional, Transaction Level Model or Bus Cycle Accurate model, Transfer Model, Cycle Accurate Model, RTL) and in SpecC (Specification Model, Processing-Element Assembly Models or IP Assembly-Model, Architecture Model, Bus Arbitration Model, Bus Functional Model, Cycle Accurate Computation model, RTL) (Blume and al. 2006).

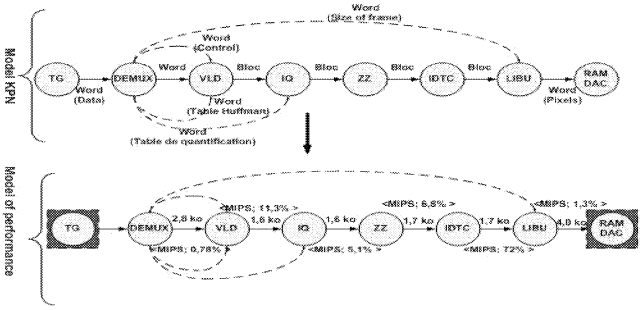


Figure 4: model of performance proposed.

We propose to add to the different abstraction levels considered in Disydent a temporal performance model. It consists in extending the Kahn process network with process execution time and the amount of exchanged data between processes. Figure 4 presents the proposed model for the MJPEG decoder. The annotations which describe the temporal performance are obtained by profiling and simulation. The tools for profiling used are Kprof, Gprof, Callgrind, Memprof. The tool for simulation used is CASS of Disydent. The model of performance can be refined on the various levels of abstractions. The objective of this model is to help the designer to quickly reach the optimal solution and to guarantee it correctly function of application.

CONCLUSION

In this paper, we presented the Disydent environment. The design flow has been tested for an MJPEG case study, to find out an optimal hardware and software implementation. However, the actual design flow lacks for a high level performance evaluation model. For this, we proposed a performance model based on the Khan Processes Network KPN witch evaluate the process execution time and the size of the exchanged data between processes. The effectiveness of this model was illustrated on the MJPEG decoder application.

REFERENCES

F. Pétrot, UPMC, Octobre 2002. « Generic Interface Modules for PI-Bus », UPMC.

F. Dumitrascu, I. Bzcivzrov, L. Pierzlisi, M. Bonaci, A.A. Jerraya, 2006. "Flexible MPSoC platform with fast interconnect exploration for optimal System performance for a specific application", Design, automation and test in Europe.

G. Kahn. 1974. "The semantics of a simple language for parallel programming". In information Processing 74, pages 471-475, Stockholm, Sweden, August 1974.

H. Blume, T. von Sydow, D. Becker, T.G. Noll, 2006. "Application of deterministic and stochastic Petri-nets for performance modeling of NoC architecture", J. Syst. Architect., doi:10.1016/j .sysarc.2006.11.001.

I. Augé, F.Pétrot, F. Donnet, P.Gomez, 2004. « Plateform based design from parallel C specification », IEEE transactions on computer-aided design of integrated circuits and systems.

I. Augé, R. K. Bawa, P. Guerrier, A. Greiner, L. Jacomme, and F. Pétrot, Aug 1997. « User guided high level synthesis» in VLSI: Integrated Systems on Silicon. IFIP Int. Conf. Very Large Scale Integration, R.Reis and L. Claensen, Eds. Gramado, Brazil: Chapman & Hall, pp. 464–475.

J.-Y. Brunel, K. A. Vissers, P. Lieverse, P. van der Wolf, W. M. Kruijtzer, W. J. M. Smits, G. Essink, E. A. de Kock, 2000. «YAPI: Application Modeling for Signal Processing Systems», dac, pp. 402-405, 37th Conference on Design Automation (DAC'00).

JPEG committee, JPEG is standardized in ISO/IEC IS 10918-1/2. [Online]. Available: http://www.jpeg.org/.

K. Popovici,, X. Guerin, F. Rousseau, P. Paolucci, A. Jarraya. 2007. "Efficient Software Development Platforms for Multimedia pplications at Different Abstraction Levels", Rapid SystemPrototypingWorkshop,Brazil.

M. Bonaciu, A. Bouchhima, W. Youssef, X. Chen, W. Cesario, A. Jerraya, 2006. "High-Level architecture exploration for MPEG4 encoder with custom parameters", IEEE.

P. Faraboschi, G. Brown, J. A. Fisher, G. Desoli, and F. Homewood. 2000. «Lx: A technology platform for customizable VLIW embedded processing», inProc. 27th Int. Symp. Computer Architecture. New York: ACM Press, pp. 203–213.

P. Quinton and V. Van Dongen, 1989. «The mapping of linear recurrence equations on regular arrays», J. VLSI Signal Process., vol. 1, no. 2, pp. 95–113, Oct.

R. Marculescu, P. Eles. 2004. « Designing Real-Time Embedded Multimedia Systems », IEEE Design & Test of Computers.

S. Vercauteren, B. Lin, and H. D. Man, Jun. 1996. «Constructing application specific heterogeneous embedded architectures from custom HW/SW applications», in Proc. Design Automation Conf., Las Vegas, NV, pp. 521–526.

COMPONENT HYBRID DYNAMIC NETS
GENERAL REPRESENTATION USING UML

S.KRABATOU
J.SAADI
siham.krabatou@gmail.com
Janah1@menara.ma
LAP (Laboratoire d'Automatique et de Productique) ENSEM Casablanca, Maroc

A.ZAKARI
zakari@univ-metz.fr
Laboratoire d'Informatique Théorique et Appliquée Université de Metz – Ile de saulcy 57045 Metz cedex 01 France

KEY WORDS
Hybrid simulation, Dynamic, Network, Bondgraph, Electrical engineering

ABSTRACT

The Component Hybrid Dynamic Nets CHDN is a labeled graphic Net used to model hybrid and continuous dynamic systems; a software tool SimRDHC using this model to simulate linear electrical and electromechanical systems was realized to validate this model.
Each model must be understood, updated, re-used and inhered by others. To achieve these goals could be through the use of a standardized documentation. This paper gives a global presentation of the structured version of the model from construction of the net to deduction of mathematical framework for simulation using UML representation.

INTRODUCTION

The CHDN is a network modelling approach suitable to simulation. It was introduced in 1992 by J.Saadi [4] [5]. This approach provides the user with simple representation and powerful tool to model complex hybrid physical systems. An advanced version ACHDN, improving the continues part of the model, is presented in [10][11].

The objective of this work is giving a general and structured presentation of the model to reach people and help them to evaluate the performance of the model. In this paper we propose an UML representation of the CHDN model. This new structuring of the model allows holding up many properties like Oriented Object architecture, hierarchical models, modular concept and reusable modelling.

The paper is organised as follow, in Section 2 we describe the basic concept of CHDN. The 3rd section presents an electrical example to illustrate the approach. Section 4 introduces the proposed structuring CHDN model using reference nodes. Section 4 shows how deducing the dynamic of hybrid system from the global CHDN net. And section 5 concludes this paper and provides some perspectives of work.

BASIC CONCEPT OF CHDN MODEL

The modeling and analysis of hybrid physical systems is a challenging area which is actually dominated by two main lines: a functional analysis based on the description of discrete state of the system (hybrid automata / ***PetriNet***), and a dynamical evolution of the system (Hybrid ***Bond Graph***, controlled differential equations). The CHDN supports the formal methods and energetic concept of modeling continuous system by a unifying graphical formalism CHDN net.

Any physical system is composed of many elements interacting with each other by exchanging energy and information and this exchange in can be represented in a graphical form. From the graph we can deduce the dynamical evolution of the system. The CHDN follows this basic concept. Furthermore, it uses component elements. The basic components support energy interpretation (resistance, inductance, capacitance, transformer and gyrator) and information is exchanged using discrete part of the model.
In upper level of hierarchy, each component has a power connector with a flux and effort variables and can have also a signal connector.
The CHDN gives a graphical representation which unifying all physical systems and hybrid concept in a same model. It allows easy deduction of mathematical framework and simulation. Examples of simulation using the software SimRDHC [7] demonstrate the power of the model.

FORMALISM OF THE ACHDN NET

The main characteristic of CHDN net is that the nodes (places, transitions and arcs) are partitioned in two groups: discrete primitive nodes that handle discrete model token (as in standard Petri nets) and continuous nodes that handle continuous dynamics. Properties of second type of nodes are:
- Each place is associated to a state variable and transition is associated to a continuous variable.
- The marker of a place can be negative (value of a continuous variable) differently to Petri Net.
- Transitions are permanently validated.
- Transition and place can have a Guard (logical condition).

Hence, in the single formalism, both discrete states and continuous variables (marker of discrete places and state variables) can be accommodated and their mutual interaction represented by Guards.

To illustrate the mutual influence between continues and discrete part of the CHDN net, we propose a simple electric example in figure1-2-3-4. Diodes and transistors represent hybrid components. They have a continuous CHDN net and a logic CHDN net and interaction is represented by logical guards.

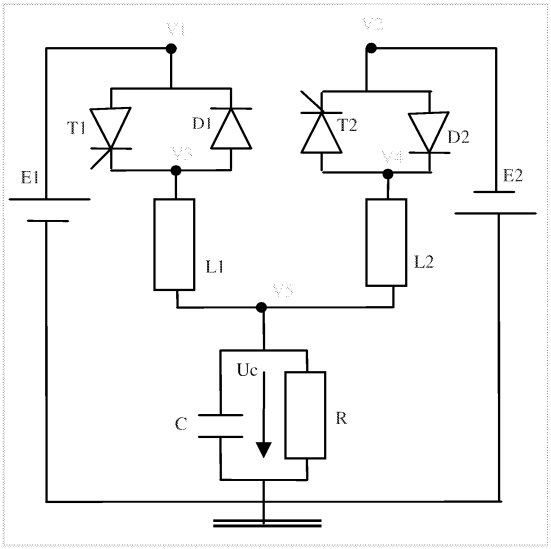


Figure 1: Electrical example

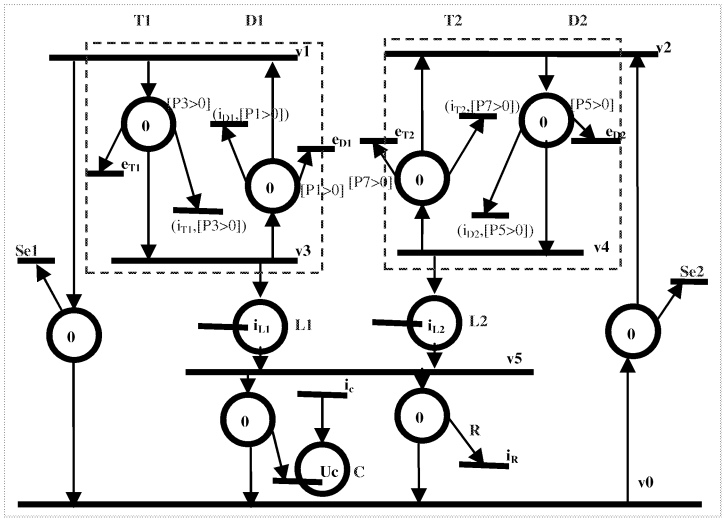


Figure 2: The effort ACHDN

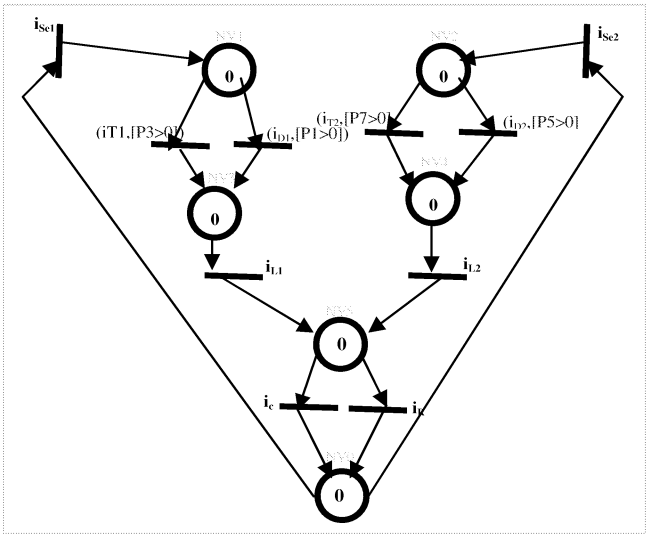


Figure 3: The flux ACHDN net

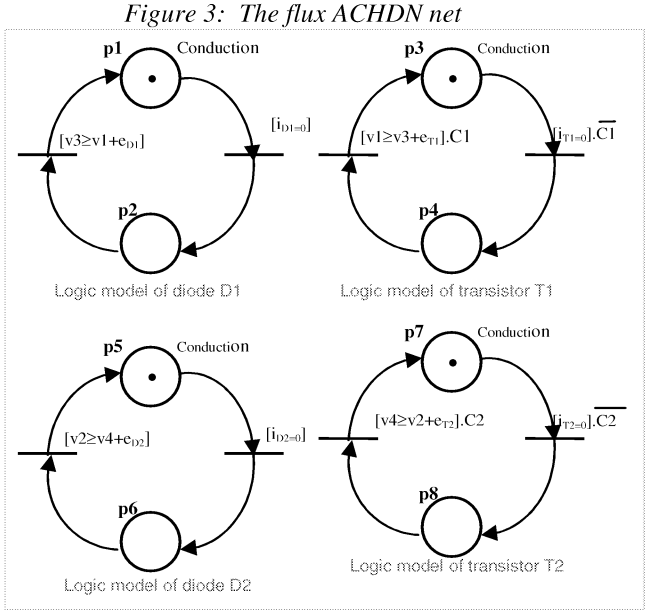


Figure 4: The logic ACHDN net

The ***effort net*** has the same topology as the system and components are clearly represented.
The ***flux net*** represents the connections between components and specially node law.
Each hybrid component (Diode transistor,…) has a continuous part represented on effort/flux net and has also a elementary ***logic net*** witch controls the states of the components.
The global CHDN net will include the three elementary nets (effort, flux and logic). So structuring the model is useful in modelling complex physical systems.

STRUCTURED CHDN MODEL USING UML

As explained in basic concept, the CHDN can support a hierarchical concept:
Level 1: basic continuous component (resistances-capacitance-inductance-transformer –gyrator...);
Level 2: basic hybrid components: are hybrid components witch have a simple continuous part;
Level 3: submodel is a connection of different components;
Level 4: model is the model of the studied system.

In other hand, every basic component has a CHDN net composed of elementary nets (logic, effort and flux net). The global CHDN net of a system is composed of component nets and their interactions. This modular concept need the introduction of a new object reference node. It represents an appearance of a node in elementary nets. The notion of reference node -as in high level Petri net HLPN[3]- is introduced in order to have only one node and one variable to refer to all time.

The reference node facilitates the interpretation of connections between physical components and help deducing differential equations of the system.

Furthermore, reference nodes help on construction of Logic functions because guards use variables from different components

A UML model presents this notion clearly. See following figure:

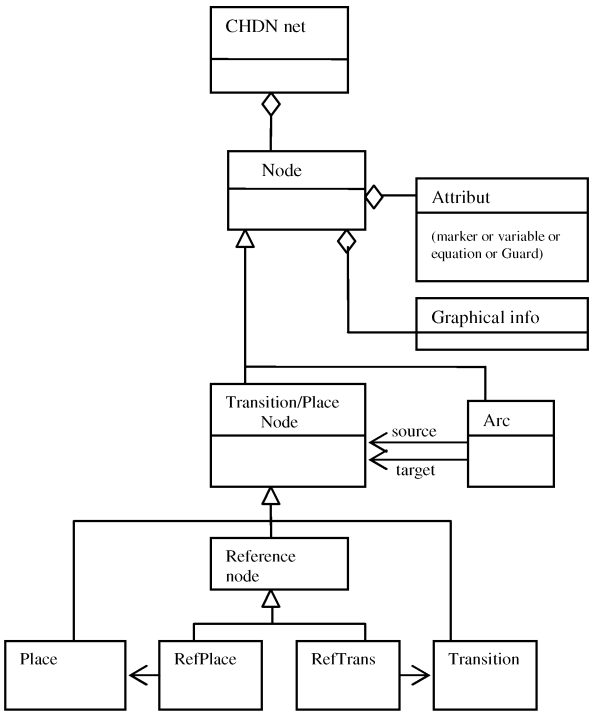


Figure5: the UML model of global CHDN net

DEDUCTION OF DYNAMIC EVOLUTION FROM THE CHDN NET

In hybrid system, the set of differential equations describing the continuous dynamic of the model changes at every state of the system. So deducing equations from the global ACHDN of a system follow this flow chart:

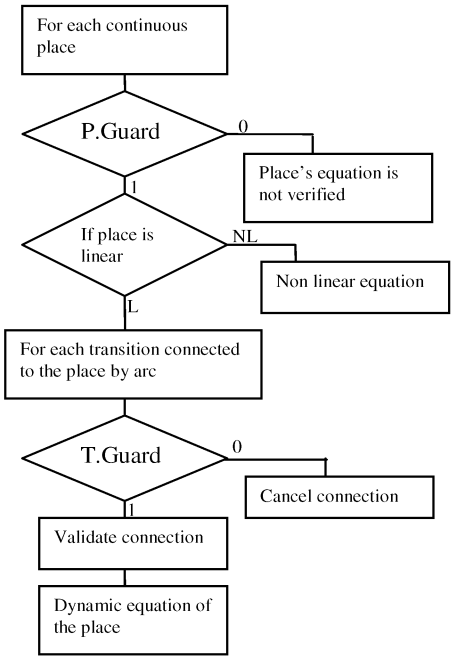


Figure7: flow chart for deduction of differential equation at one state of the system

From each place of the net we can deduce an equation which can change by changing the state or to be invalid. Resolution of differential equations is done at every step of simulation with checking the state of the hybrid system.

CONCLUSION

In this paper, we have presented the principles and concepts of the CHDN model. A simple electrical example is illustrated to help on constructing the CHDN net. We provide also UML analysis to give more details in structuring model.

The perspectives of the work are the definition of models libraries enabling the modelling of hybrid component and a simulation software using ACHDN model.

REFERENCES

[1] Cellier, E.F. 1991. "Continuous system modelling", Springer-Verlag.

[2] D. Karnopp, R. Rosenberg. 1975. "System dynamics: A unified approach," willey-Interscience publication.

[3] Ekkart kindler. 2004. "Highlevel Petri Nets-transfer format" Working Draft of the International Standard ISO/IEC 15909 Part 2 University of Padeborn.

[4] J. Saadi. 1994. "Diagnostic et modélisation de défauts au sein d'une association convertisseur - machine", PhD Es-sciences, University Mohamed V, Morocco.

[5 J. Saadi, T. Bennani, and H. Alla. 1997. "Component hybrid dynamic nets." IFAC/IFIP Conference on Management and Control of Production and Logistics, pages 211-216.

[6] M. Chami, A. Djerdir, J. Saadi, A. Miraoui. 2004. "The Use of the CHDN to model a Permanent Magnet Synchronous Motor powered by Ultracapacitors: Electrical Vehicle Application," CD ROM Symposium IEEE Vehicle Power and Propulsion, WP4-3, Paris.

[7] M. Chami. 2005. " modélisation et simulation des systèmes multi-physiques à l'aide des réseaux dynamiques hybrides à composants :application à la conversion d'énergie et au transport électrique terrestre", PhD Es-sciences, University of Franche-Comté and University of Technology Belfort-Montbéliard France and University Hassan II Ain Chok Morocco.

[8] P.E. Wellstead. 1979. "Introduction to physical system modelling", Academic Press.

[9] R. David and H. Alla. 1987. "Continuous Petri nets", 8th European conference on theory and application of Petri nets, Saragossa (E) pp 275-294.

[10] S.Krabatou, J.Saadi. 2005. "Advanced Component Hybrid Dynamic Net (ACHDN)" AMSE Conference in France.

[11] S.Krabatou, J.Saadi ,A.zakari. 2006. "Thermal Advanced Hybrid Dynamic Net " ISCCSP06.

Simulation of integrated control system for climate comfort

Mariusz Nowak
Andrzej Urbaniak

Institute of Computing Science,
Poznan University of Technology,
Piotrowo 2,
60-965 Poznan, Poland
Emails: Mariusz.Nowak@cs.put.poznan.pl, Andrzej.Urbaniak@cs.put.poznan.pl

KEYWORDS: air conditioning, indexes of thermal comfort, fuzzy logic control, model predictive control.

ABSTRACT

Nowadays, in modern industrial society people spend most of their lifetime in artificial environment – at home, at workplace, in shops. The air-conditioners can be seen in buildings of office type, such as concert hall, conference room, lecture hall, operating room in hospitals. The task of air-conditioner is to assure the feeling of microclimatic comfort in a room. The air-conditioner must be the dependable and economical device. However, the feeling of microclimatic comfort is quite a relative notion, depending on very individual feelings. The microclimatic comfort consist largely in thermal comfort which is influenced by many parameters [2,3]. In the paper there is presented the mathematical model of thermal comfort. There is also described the hierarchical structure of climate comfort's control. Additionally, there is presented the use of fuzzy control algorithms in direct control layer and model predictive control algorithms in supervisory control layer.

1. INTRODUCTION

Thermal sensation of a human being is related mainly to thermal balance of its body. The balance is influenced by two groups of factors: personal and physical ones. Activity level and clothing thermal insulation belong to the group of personal factors. Environmental parameters, such as air temperature, mean radiant temperature, air velocity, air humidity compose physical factors. The physical factors measuring thermal sensation of a body can be predicted by calculating the PMV index. The PMV (Predicted Mean Vote) is used as thermal comfort indicator that shows mean thermal sensation vote for a group of people. The PMV is defined by six variables: air temperature, humidity, air velocity, mean radiant temperature, clothing insulation and human activity. The second index is PPD (Predicted Percentage of Dissatisfied), which is obtained from the PMV index. PPD index provides information on thermal discomfort by predicting the percentage of satisfied or dissatisfied people. Additionally, the unwanted local cooling or heating of the body may cause thermal discomfort. The common local thermal discomfort is draught, defined as local cooling of the body caused by air movement. The next index is DR (draught rating) that predicts the percentage of people dissatisfied due to a draught. PMV, PPD and DR are synthetic indexes [2,3,4,7].

2. THERMAL COMFORT

Thermal comfort is defined in British Standard EN ISO 7730 as "that condition of mind which expresses satisfaction with the thermal environment" [1]. It means that the term describes one's psychological state of mind and is usually referred to whether someone feels too hot or too cold.

The major objective is to create environment where each person can live in thermal comfort. Obviously, the thermal comfort is a subjective sensation and satisfying everyone in a given environment may be difficult. Therefore, the realistic aim is to create environment, which satisfies maximum possible percentage of people in a given room, taken as a group.

There are two conditions to be performed to maintain the thermal comfort. First one is actual combination of skin temperature and body's core temperature providing a sensation of thermal neutrality. Second one is fulfilment of body's energy balance. Heat produced by metabolism should be equal to amount of heat lost from the body by conduction, convection, radiation, evaporation and respiration. These two conditions are combined in a single equation called the comfort equation. The comfort equation can be written as (1):

$$M - W = H + E_c + C_{res} + E_{res} \qquad (1)$$

where:

- M - metabolism [W/m^2],
- W - performed mechanical work [W/m^2],
- H - body heat loss by convection, radiation and conduction (dry heat loss) [W/m^2],
- E_c - evaporative heat exchange on skin when subject experiences a sensation of thermal neutrality [W/m^2],
- C_{res} - respiratory convective heat exchange [W/m^2],
- E_{res} - respiratory evaporative heat exchange [W/m^2].

3. THERMAL COMFORT INDEXES

It was Fanger who described PMV thermal comfort degree index first presenting thermal comfort degree as a feeling of most people [2]. The PMV index is shown in Table 1.

The PMV index is based on heat balance of human body and is given by the empirical equation (2):

$$PMV = (0.028 + 0.303e^{-0.036M})\left[(M - W) - H - E_c - C_{res} - E_{res}\right] \quad (2)$$

where E is actual evaporative heat exchange on subject's skin. The subject is in thermal balance when the internal heat production in the body is equal to the loss of heat in the environment i.e. $E=E_c$.

The PMV index predicts the mean value of votes of a large group on the following seven–point thermal sensation scale:

Table 1. PMV index

Feeling of hot	Cold	Cooler	cool	comfort	warm	warmer	Hot
PMV	-3	-2	-1	0	1	2	3

From the equation (2) the PMV can be calculated for different combinations of metabolic rate, clothing, air temperature, mean radiant temperature, air velocity and air humidity. The equations for t_{cl} and h_c may be solved by iteration. Next index is PPD: it gives the expected number in per cent for a group of people who are not satisfied with the thermal climate. PPD index is given by the following equation (3):

$$PPD = 100 - 95 \cdot e^{-\left(0.03553 \cdot PMV^4 + 0.2179 \cdot PMV^2\right)} \quad (3)$$

To predict the percentage of people dissatisfied due to a draught, EN ISO 7730 introduces the draught rating index DR, which is given by the following equation [1]:

$$DR = \left(34 - t_a\right)\left(v - 0.05\right)^{0.62}\left(0.37 \cdot T_u \cdot v + 3.14\right) \quad (4)$$

where: T_u - turbulence intensity.
t_a - air temperature,
v - air velocity.

4. HIERARCHICAL STRUCTURE OF CONTROL

The goals of control have got a different nature. They the most often have an economic character [8,9]. Realization of a principal goal depends on realization of fragmentary goals that guarantees the safety of process. The process assuring climatic comfort is a complex one that makes the control very difficult. The solution is use of hierarchic control. Block scheme of hierarchical structure is presented in Fig. 1.

The lowest layer – direct control layer, assures the safety of the controlled process. This is the only layer with direct access to the controlled system. Algorithms in this layer have to be dependable and simple. Usually, the classic PID algorithms, direct digital algorithms or fuzzy logic algorithms are used. The higher layer is the supervisory control one. The main tasks of this layer is the evaluation of set-values in progress for controllers. The algorithms of control in this layer be supported on the model of process. The algorithms of predictive control be practical in supervisory control layer. Next layer – the optimization one is over the supervisory control layer. In this layer the optimum set-values is calculate for controllers of supervisory control and direct control layers. The next - planning layer determines the goal and limitation for optimization layer.

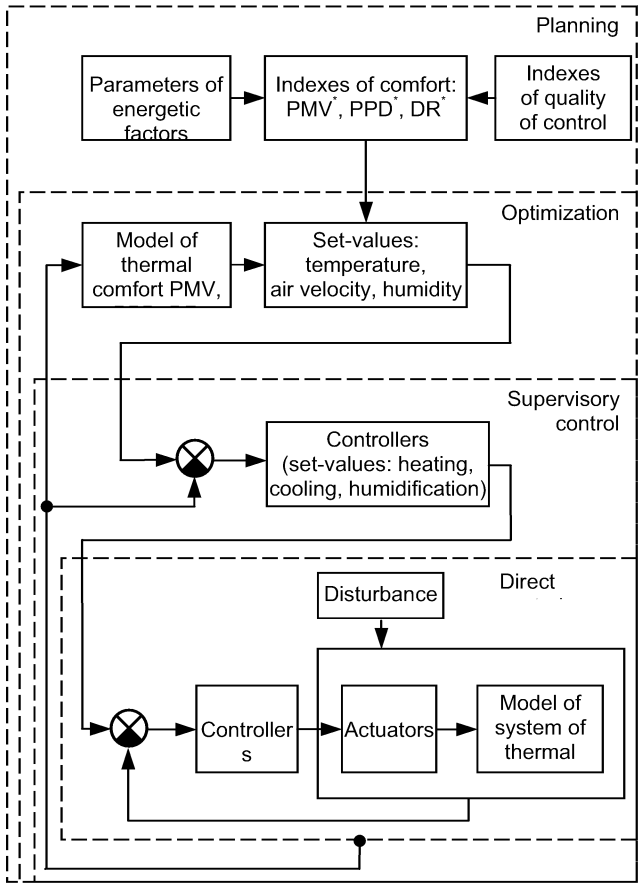


Fig 1. Hierarchical structure of climatic comfort control

5. ALGORITHMS OF CLIMATIC COMFORT CONTROL

There are fuzzy logic controllers used in the direct control layer. The approach based on classical PI or PID controllers is not sufficient for climatic comfort control processes. Therefore, there is fuzzy approach for these processes implemented. The fuzzy controller consists of the following components:

- fuzzyfier,
- inference module,
- defuzzyfier.

The basic structure of a Fuzzy Logic Control (FLC) system is shown in Fig. 2.

Fuzzification is the process of mapping an input to the FLC into fuzzy membership values in various input universes discourse. Decision need to be made regarding:

- number of inputs,
- size of universes of discourse,
- number and shape of fuzzy sets.

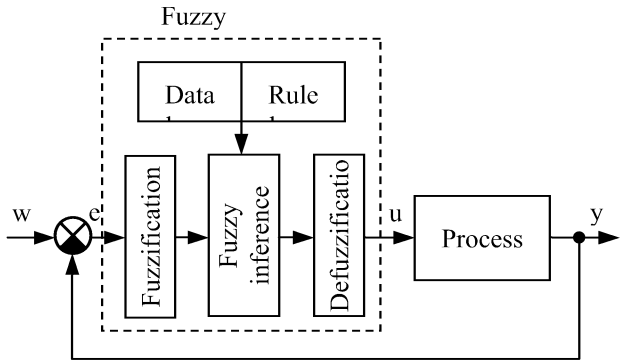
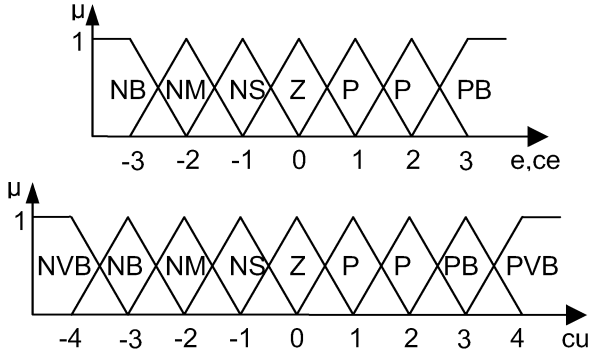


Fig. 2. Fuzzy logic control system

The input signals for fuzzy controllers are: error (e) and error change (ce). The output signal for PI fuzzy controllers is change of control signal (cu). Fuzzy intervals for the input and output signals are shown in Fig. 3.



N-negative, Z-zero, P-positive, VB-very big, B-big, M-medium, S-small

Fig 3. Fuzzy interval membership functions for e, ce and cu.

The fuzzy rule base consists of a set of antecedent-consequent linguistic rules in the following form:

IF *e* is PS and *ce* is NS THEN *cu* is PS

This style of fuzzy conditional statement is often called the Mamdani type rule.

Fuzzy inference exists to translate all expressions to fuzzy language notation.

Defuzzification is the procedure for mapping from a set of inferred fuzzy control signal, contained within a fuzzy output window, to a non-fuzzy control signal. The calculation method of this value is the well known defuzzification technique [9].

In the system of climatic comfort's control the example rules have the following form:

IF *temperature* is high and *air speed* is not very fast
THEN *pressure* is slightly decreased

In this example, two input variables: *temperature* and *air speed* have values defined as fuzzy sets. The output variable, *pressure,* is also defined by the fuzzy set with values like: *static*, *slightly increased*, *slightly decreased*, etc.

In supervisory control layer the predictive control was used. Model Predictive Control (MPC) is based on iterative, finite horizon optimization of a plant model.

At time *t* current plant state is sampled and control strategy for cost minimisation is computed for a relatively short time horizon in the future. Specifically, an online calculation is used to explore state trajectories emanating from current state and to find a cost-minimizing control strategy until *t+ time step*. Only the first step of the control strategy is implemented, than the plant state is sampled again and calculations repeat starting from the new current state.

The prediction horizon keeps being shifted forward and for this reason MPC is also called receding horizon control. Although this approach is not optimal, it has been given very good results in practice.

Model Predictive Control is a multivariable control algorithm that uses:

- an internal dynamic model of the process,
- a history of control moves,
- a cost optimization function J over the prediction horizon to calculate the optimum for control moves.

The cost optimization function is given by:

$$J(k)=\sum_{p=N_1}^{N}\left\|w(k+p\,|\,k)-y(k+p\,|\,k)\right\|^2+\lambda\sum_{p=0}^{N_u-1}\left\|\Delta u(k+p\,|\,k)\right\|^2 \quad (5)$$

where:

J(k) - cost optimization function,
w(k+p|k) - set-value evaluated for k+p moment in k moment,
y(k+p|k) - regulated value evaluated for k+p moment in k moment,
Δu(k+p|k) - increment of control value evaluated for k+p moment in k moment,
λ - damping factor,
N - horizon of prediction,
N_1 - start time of calculating the criterion function,
N_u - horizon of control.

The predictive controller requires the model of object control. The Matlab System Identification Toolbox is used for calculation model of control object [7].

In the optimisation layer there was the costs criterion implemented. This criterion was expressed as the sum of the costs of air preparation and losses of costs focusing on non-fulfilment of expected microclimate conditions. The

results of this control layer are used to obtain set values for controllers utilizing in the lower control layers.

The greatest one is the planning layer. In this layer the production goals are defined in the longer horizon.

This layer is incorporate in the Building Management System (BMS) which integrates the software responsible for intelligent object management as a whole. The operation mode in this layer utilizes the forecast methods based on the specialized methods [8,9].

6. SIMULATION RESULTS

The mathematical model of thermal comfort was implemented in Matlab. There were being realised two simulations using different algorithms.

In the first one classical algorithms were used and intelligent algorithms in the second one. In the direct control layer there were fuzzy algorithms and in the supervisory control layer the predictive ones.

For example, there are shown the courses of variability of coefficients' quality of thermal comfort: courses of variability of indexes: PMV and PPD are presented in Figure 4 and Figure 5.

There are the courses of power heater (Pn) and power cooler (Pc) presented in Fig. 6 and Fig. 7.

In all the figures the dashed lines present the result of classical algorithms' effect – PID and solid lines present the result of intelligent algorithms' effect – fuzzy logic and predictive algorithms.

The obtained results confirm the effectiveness while implementing intelligent algorithms comparing with the classical ones. This effectiveness focuses in lower costs of HVAC system exploitation which is the main cost within the whole building exploitation costs.

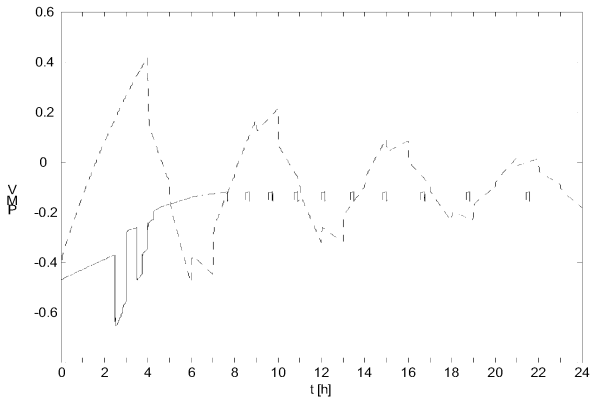


Fig 4. PMV=function(time)

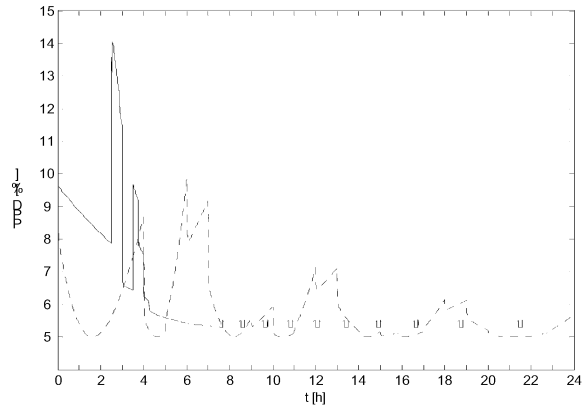


Fig 5. PPD=function(time)

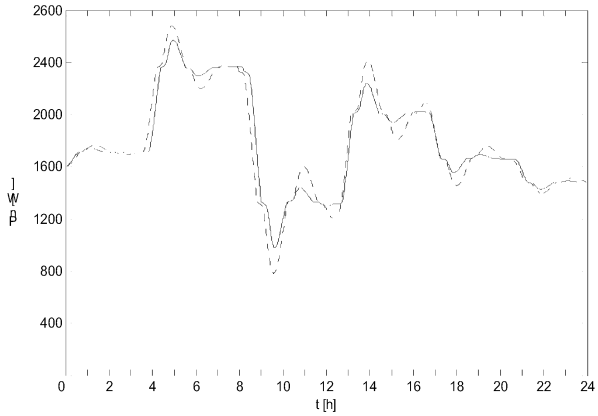


Fig 6. Pn=function(time)

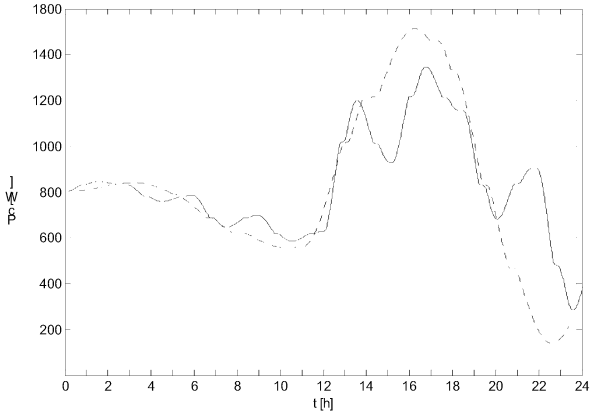


Fig 7. Pc=function(time)

7. CONCLUSIONS

There were two approaches compared in the simulations. The first approach is based on the classical algorithms while the second presents the hierarchical structure for control of climatic comfort. The approach based on the intelligent algorithms is by no means better than the first one. The indexes PMV and PPD are more stable. And what is more, the courses of changes of heater power and cooler power show lower costs of control using the second approach.

7. REFERENCES:

[1] EN ISO 7730:1994, *Moderate thermal environments – Determination of the PMV and PPD indices and specification of the conditions for thermal comfort*

[2] Fanger, P.O., *Thermal Comfort*, McGraw-Hill, New York, USA 1973

[3] Hampreys M.A., Nicol J.F., *Understanding the adaptive approach of thermal comfort*, ASHRAE Transations, 1998, Part 1B

[4] Jones W. P., *Air Conditioning*, Wydawnictwo „Arkady", Warszawa 2001 (in polish)

[5] Nowak M., Urbaniak A., *Computer simulation of the influence of physical parameters on thermal comfort*, International Carpathian Control Conference ICCC'2004, Poland, Zakopane, 25-28 May 2004, ISBN 83-89772-00-0, pp. 481-484

[6] Nowak M., Urbaniak A., *Computer modelling and simulation of the influence of physical parameters on thermal comfort*, in: The 2005 European Simulation and Modelling Conference - Proceedings, J. Manuel Feliz Teixeira, A. E. Carvalho-Brito (ed.), Pub. EUROSIS-ETI, ISBN 90-77381-22-8, Porto−Portugal 2005, pp. 445-447

[7] Nowak M, Zakrzewski P., *Employment of Matlab packet to the simulation the chosen processes influencing on the environment quality*, Computer in environment protection - Scientific & Technical Conference, Poland, Poznań-Wągrowiec 2003, ISBN 83-9142252-4-8, pp. 27-45 (in polish)

[8] Olsson G., Piani G., *Computer systems in automation*, Prentice-Hall, London – New York 1992

[9] Tatjewski P., *Advanced object's control – structures and algorithms*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002 (in polish)

AUTORS BIOGRAPHIES

Mariusz NOWAK was born in Poland and went to the Poznan University of Technology (control engineering). He obtained the MSc degree in 1999. From October 1999 he is an Assistant at the Institute of Computing Science of the Poznań University of Technology. He research interest include: computer simulation, intelligent control system, computer control systems for environmental engineering, intelligent building systems. Now he is finishing your PhD dissertation concerning comfort climate comfort.

Andrzej URBANIAK was born in Poland and went to the Poznan University of Technology (control engineering) and Poznan University of A. Mickiewicz (mathematics). He obtained the PhD degree in 1979. From 1990 he is a professor of Institute of Computing Science and Deputy Rector of The State Higher Vocational School in Gniezno. He is author or co-author of 5 books and over 250 papers concerning the computer control systems and application of computer science in environmental engineering.

AUTHOR LISTING

AUTHOR LISTING

Abad F. **514**
Adamko N. **364**
Agha A. **89**
Ahvenlampi T. **227**
Albert V. **29**
Alfonseca M. **458**
Alpan G. **178**
Ammari A.C. **600**
Anagnostopoulos D. **296**
Anand A. **172**
Andrade A. **389**
Auberlet J.-M. **371**
Aufenanger M. **499**

Balasko A. **569**
Barbero A. **458**
Bártolo P.J. **235**
Batanov D.N. **163**
Benaissa K. **345**
Benjelloun-Dabaghi Z. . **142**
Berger F. **582**
Bernon C. **432**
Bertelle C. **262/306/309/315**
Besga J.M. **348**
Beurton-Aimar M. **537**
Biro M. **555**
Bleimann U. **452**
Bourhis M. **447**
Bouskela D. **137**
Brito A.C. **107**
Bruneel H. **214**

Calvez B. **415**
Camahort E. **514**
Camilleri L. **191**
Cannella S. **95**
Capra L. **155**
Caris A. **163/283**
Castellano E. **348**
Ceska M. **170**
Chappin E. **356**
Ciancimino E. **95**
Claeys D. **214**
Clément E. **309**
Conley W. **287**
Courvoisier M. **29**

Dagis S. **336**
Dalakas V. **296**
Däne B. **582**
Dangelmaier W. **499**
de Lara J. **458**
Devocelle M. **409**
Diep D. **345**
Dijkema G. **356**
Dormido S. **147**
Duchamp G.H.E. **262/306**
Dumond Y. **547**
Dupleac D. **555**
Dutot A. **259/267**
Dvornik J. **65**
Dvornik S. **65**

Fedorèák D. **331**
Feliz-Teixeira J.M. **68**
Fengler W. **582**
Fischer M. **499**
Fister S. **493**
Fortino G. **437**

Galuszka A. **321**
Garro A. **437**
Gerzson M. **81**
Ghnemat R. **306**
Ginoux J.-M. **255**
Gleizes M.-P. **432**
Gonzalez-Hidalgo M. **74**
González-Rodríguez M.S. .. **458**
Grandt M. **57**
Greco S. **475**
Grunde-Zeiferts U. **533**
Guinand F. **259**

Hacia L. **595**
Harbegue H. **600**
Hatzi O. **296**
Heinzl R. **483/488/493/506**
Hermann G. **569**
Hidalgo J.L. **514**
Huber D. **499**
Hutzler G. **415**

Jaff L. **262**
Janssens G.K. **163/249/283**
Jay V. **309**
Jelic-Mrcelic G. **37**
Jemai A. **600**

Kacsuk P. **569**
Kacsukné Bruckner L. **569**

AUTHOR LISTING

Kanold M. **452**
Kaproń H. **92**
Kaproń T. **92**
Karlowatz G. **220**
Kausch B. **57**
Kavička A. **364**
Kindler E. **20/124**
Klima V. **364**
Kortenjan M. **499**
Kosina H. **220**
Krabatou S. **603**
Kreider M. **452**
Kreslikova J. **205**
Krivy I. **124**
Krokowski J. **499**
Kulkarni A. **560**
Kvetonova S. **205**

l'Ecuyer P. **5/48**
Laevens K. **214**
Laroque C. **499**
Lawrence A. **525**
Lawrence P. **519/525**
Leitold A. **81**
Lemouzy S. **432**
Lencse G. **291/574**
Lukszo Z. **356**

Mach G. **483/488/493**
Machado J. **394**
Macharis C. **283**
Mahboub K. **309**
Makackas D. **427**
Marlewski A. **595**
Marreiros G. **394**
Martinez F. **147**
Martín-Villalba C. **147**
Mascillaro S. **437**
Matias J.M. **235**
Mazat J.P. **537**
Mir A. **74**
Moalla M. **600**
Morais H. **389**
Mosler H.-J. **373**
Mozga I. **533**
Mueck B. **499**
Muka L. **291/574**
Munitic A. **37**
Murphy J.T. **409**

Najafi M. **137/142**
Nassehi V. **560**
Neves J. **394**
Nicolau-Bestard G. **74**
Nikolaidou M. **296**
Nketsa A. **29**
Novais P. **394**
Novosad P. **170**
Nowak M. **606**

Oliver I. **272**
Olivier D. **259**
Pacholczyk M. **542**
Pacholik A. **582**
Palopoli F. **475**
Palosaari M. **227**
Paludetto M. **29**
Parisey N. **537**
Paulauskaite-Taraseviciene A. ... **427**
Pigné Y. **259**
Pogoreutz C. **493**
Pohlmann W. **112**
Praca I. **389**
Pranevicius H. **427**
Prisecaru I. **555**

Qaddoum K. **315**

Rabenberger M. **112**
Rajpal N. **172**
Ramaekers K. **163/249**
Ramos C. **389/394**
Rarità L. **230**
Reddy B.V.R. **172**
Remondino M. **402**
Rodin V. **447**
Roest G.B. **381**
Rossetto B. **255**
Russo W. **437**

Sá Marques P. **107**
Saadi J. **603**
Sabater-Mir J. **467**
Salemi R. **560**
Schlick C. **57**
Schwaha P. **483/488/493/506**
Selberherr S. **220/483/488/493/506**
Sitte R. **13**
Sklenar J. **119**

AUTHOR LISTING

Skrzypczyk K. **303**
Sliskovic M. **37**
Smiri K. **600**
Sosnowski Z.A. **326**
Sroczan E.M. **199**
Stalidzans E. **533**
Starosolski Z. **542**
Stengel I. **452**
Stimpfl F. **506**
Strelen J.-C. **40**
Stynes J. **452**
Sudars S. **533**
Sverdlov V. **220**
Szirbik N.B. **381**

Tackenberg S. **57**
Tanguy A. **129**
Tawalbeh N. **89**
Thabtah F. **315**
Thomas S. **296**
Tobias R. **373**
Tranouez P. **267**
Trapel E. **230**
Tropper C. **586**
Tuffin B. **48**
Tuncel G. **178**

Urbaniak A. **606**
Uribetxeberria J. **348**
Urquia A. **147**

Vale Z. **389**
van Dam K. **356**
Van Loock R. **210**
Van Oudheusden D. **210**
Van Utterbeeck F. **210**
Villatoro D. **467**
Vivo R. **514**
Vondrák I. **331**

Walraevens J. **214**
Walshe R. **409**
Wang J. **586**
Würzebesser C. **373**

Zakari A. **603**
Zugasti I. **348**