MODELLING AND SIMULATION 2008

THE EUROPEAN SIMULATION
AND
MODELLING CONFERENCE
2008

ESM®'2008

EDITED BY
Cyrille Bertelle
And
Aladdin Ayesh

OCTOBER 27-29, 2008
LE HAVRE, FRANCE

A Publication of EUROSIS-ETI
The European Simulation and Modelling Conference 2008

LE HAVRE, FRANCE

OCTOBER 27-29, 2008

Organised by
ETI- The European Technology Institute
Sponsored by
EUROYSIS, The European Simulation Society

Co-Sponsored by

Ghent University

l'Université du Havre

and

LITIS

Hosted by

L’Université du Havre
Le Havre, France
EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

General Conference Chair
Cyrille Bertelle, Universite du Havre, le Havre, France

General Conference Co-Chair
Alladin Ayesh, de Montfort University, Leicester, United Kingdom

Past Conference Chair
Jaroslav Sklenar, University of Malta. Msida, Malta

Journal Publication Chair
Laurence Tianruo Yang, St. Francis Xavier University, Canada

Local Committee Chairs
Cyrille Bertelle, Universite du Havre, le Havre, France
Damien Olivier, Universite du Havre, le Havre, France
Frédéric Guinand, Universite du Havre, le Havre, France
M.A. Aziz-Alaoui, Universite du Havre, le Havre, France
Stefan Balev, Universite du Havre, le Havre, France
Antoine Dutot, Universite du Havre, le Havre, France
Véronique Jay, Universite du Havre, le Havre, France

INTERNATIONAL PROGRAMME COMMITTEE

Methodology and Tools
Thomas Hanne, Fraunhofer -ITWM, Kaiserslautern, Germany
Bjorn Johansson, Chalmers Univ. of Techn. Gothenburg, Sweden
Abder Koukam, Univ.de Technologie de Belfort, Belfort, France
Moreno Marzolla, Universita di Venezia, Mestre, Italy
Roberto Revetria, University of Genova, Italy
Bert van Beek, Eindhoven University of Technology, The Netherlands
Abdou Zakari, LITA, Universite de Metz, France

Simulation and Artificial Intelligence
Helder Coelho, Fac Ciencias, Lisbon, Portugal
Paulo Cortez, University of Minho, Guimareas, Portugal
Adam Galuszka, Silesian Technical University, Glwice, Poland
Martin Hruby, Brno University of Technology, Brno, Czech Republic
Vladimir Janousek, Brno University of Technology, Brno, Czech Republic
Esko Juuso, University of Oulu, Finland
Jose Neves, Universidade do Minho, Braga, Portugal
Wolfgang Kreutzer, Univ. of Canterbury, Christchurch, New Zealand
Damien Olivier, Universite du Havre, France
Maja Pantic, Imperial College, London, United Kingdom
Leon Rothkrantz, TU Delft, The Netherlands
Franciszek Seredyński, Polish Acad.of Science, Warsaw, Poland
Jim Torresen, University of Oslo, Norway
Frantisek Zboril, Brno University of Technology, Brno, Czech Republic
Morched Zeghal, Nat. Res.Council Canada, Ottawa, Canada
INTERNATIONAL PROGRAMME COMMITTEE

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
Jari Porras, Lappeenranta University of Technology, Finland
Simon See, Sun Microsystems Inc, Singapore
Pierre Siron, ONERA, Toulouse, France
Behrouz Zarei, Sharif University of Technology, Tehran, Iran

Simulation in Education and Graphics Visualization
Magy Seif El-Nasr, Penn State University, Evanston, USA
Jan Lemeire, VUB, Brussels, Belgium
Qingping Lin, Nanyang Technological University, Singapore
Marco Roccetti, University of Bologna, Italy

Simulation in Environment, Ecology, Biology and Medicine
Eduardo Ayesa, CEIT, San Sebastian, Spain
Cyrille Bertelle, LIH, Le Havre, France
Thiavd Chaussalet, University of Westminster, London, UK
Vahid Nassehi, Loughborough University, United Kingdom
Laurent Perrochon, 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
Roland Haas, DaimlerChrysler Research and Technology, Bangalore, India
Clemens Heitzinger, Purdue University, West Lafayette, USA
Francesco Moscato, Seconda Univ.of Naples, Aversa, Italy
Panajotis Katsaros, Aristotle University, Thessaloniki, Greece
Eva M. Rubio, UNED, Madrid, Spain

Web Based Simulation
Dimosthenis Anagnostopoulos, Harokopion University of Athens, Greece
Manuel Alfonseca, Universidad Autonoma de Madrid, Spain
Ammar Al-Khani, VTT Processes, Espoo, Finland
Jose Barata, New University of Lisbon, Portugal
Lenin Lemus, UPV, Valencia, Spain
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
INTERNATIONAL PROGRAMME COMMITTEE

Agent Based Simulation
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 Porto, Braga, Portugal
Jan-Torsten Milde, FH Fulda, Germany
Isabel Praca, Ist. Superior do Porto, Portugal
Marco Remondino, University of Turin, Italy
Agosthino Rosa, Technical University Lisbon, Portugal

Simulation with Petri Nets
Pascal Berruet, Universite Bretagne Sud, Lorient, France
Carmen Bobeanu, Ghent University, Belgium
Mauro Iacono, University of Naples II, Italy
Juan de Lara, Univ. Autonoma de Madrid, Spain
Hammid Demmou, LAAS CNRS, Toulouse, France
Olivier Grunder, UTBM, Belfort, France
Guenther 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
Jean-Claude Pascal, 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, Corte, France
Fernando Tricas, Universidad de Zaragoza, Spain

Fluid Flow Simulation
Diganta Bhusan Das, Loughborough University, United Kingdom
H.A.Nour Eldin, University of Wuppertal, Germany
Markus Fiedler, Blekinge Institute of Technology, Sweden
Hai Xiang Lin, TU Delft, The Netherlands

Models and simulations for Emergency and Risk Management
Track Chair: Damien Olivier, LITIS, UFR Sciences et Techniques, Le Havre, France

IPC
A. Ayesh, De Montfort University, Leicester, United Kingdom
E. Babkin, Higher School of Economics (Nizhny Novgorod), Russia
E. Daudé, Université de Rouen, Rouen, France
D. Lefebvre, Université du Havre, Le Havre, France
D. Olivier, Université du Havre, Le Havre, France
G. Pianchette, IMDR, France
D. Provitolo, Université de Franche-Comté et de Bourgogne, France
INTERNATIONAL PROGRAMME COMMITTEE

Complex Systems: Historical & Dynamical point of view
Track Chair: Jean-Marc Ginoux, IUT de Toulon, La Garde, France
IPC
Cyrille Bertelle, University of Le Havre, Le Havre, France
Aziz Alaoui, University of Le Havre, Le Havre, France
Bruno Rossetto, Protee, Toulon, France
Jean-Marc Ginoux, Protee, Toulon, France
Gérard H.E. Duchamp, Paris 13, France
Michel Cotsaftis, ECE, Paris, France

Emotional dimensions of cognition and their modelling in problem-solving, learning and decision-making
Chairs: Aladdin Ayesh, De Montfort University, Leicester, United Kingdom
Sophie Baudic, INSERM, Boulognes, France
IPC
Habib Abdulrab, Rouen University, Rouen, France
Aladdin Ayesh, De Montfort University, Leicester, United Kingdom
Cyrille Bertelle, Le Havre University, Le Havre, France
Gérard H.E. Duchamp, Paris 13 University, Paris, France
Véronique Jay, Le Havre University, Le Havre, France
Damien Olivier, Le Havre University, Le Havre, France
Pierrick Tranouez, Rouen University, Rouen, France

Artificial Society
Chairs: Aladdin Ayesh, De Montfort University, Leicester, United Kingdom
Cyrille Bertelle, Le Havre University, Le Havre, France
IPC
Aladdin Ayesh, De Montfort University, Leicester, United Kingdom
Edward Babkin,
Cyrille Bertelle, Universite du Havre, Le Havre, France.
Laszlo Gulyas
Eric Daudé
Bernard Elissalde

Simulation, Modelling and Data Management in Real-Time Systems (SMDM-RTS'2008)
Chairs
L. Amantone, LITIS, University of Le Havre, Le Havre, France
J.-P. Babau, INSA, Lyon, France
L. Baccouche, RIADI-GDL, University of Tunis, Tunisia
A. Bensrhar, LITIS, INSA, Rouen, France
R. Bouaziz, MIRACL, Université de Sfax, Tunisia
Ch. COLLET, LIG, Grenoble, France
F. Cottet, LSI, ENSMA, Poitiers, France
J. Darmond, ERIC, Lyon 2, France
C. Duvatelle, LITIS, University of Le Havre, France
F. Gargouri, REGIM, University of Sfax, Tunisia
A. Hammad, LIFC, University of Franche-Comte, France
M. Mainguenaud, LITIS, INSA, Rouen, France
Z. Mammeri, IRIT, Université of Toulouse, France
N. Navet, LORIA, University of Nancy, France
P. Richard, LISI, ENSMA, Poitiers, France
B. Sadeg, LITIS, University of Le Havre, France
Y.-Q. Song, LORIA, University of Nancy, France
C. Soule-Dupuy, IRIT, University of Toulouse, France
K. Yetongnon, IEE, University of Dijon, France
EUROPEAN SIMULATION AND MODELLING CONFERENCE 2008
Dear participants,

It is our pleasure, to welcome you to the 22nd annual European Simulation and Modelling Conference ESM’2008, organized by EUROSIS. This international European conference which concerns itself with the research activities on modelling and simulation, is being held at LITIS at the University of Le Havre, France. LITIS is the biggest research centre in Haute-Normandy, dealing with the Sciences and Technologies for Information and Communication.

Modelling and simulation are of major interest for both academic activities and industrial applications. The conference contributions show this cross-fertilization in a major way, containing papers on innovation in methodology (like the role of emergence and metaknowledge in simulation) and various papers on industrial applications.

Even if Mathematics and Computer Science are generally described as the Sciences of Modelling, we decided this year to highlight the pluridisciplinary approaches and the benefit of the mix between various thematic descriptions and knowledge with the modelling processes. From this point of view, the Science of Complexity proposes some new considerations and allows to unify scientific purposes from many disciplines, on similar concepts and similar scientific perceptions.

I would like to thank the keynote and invited speakers who contribute to this innovative and pluridisciplinary approach: Denise Pumain from UMR Géographie-Cités, Paris (France); Aladdin Ayesh from IMRCC Group, De Montfort (U.K.), Michel Cotsaftis from LACSC-ECE, Paris (France), Laszlo Gulyas from Lorand Eotvos University, Budapest (Hungary), Jean-Pierre Müller from Cirad, Montpellier (France) and Thierry Faure et al. from Cemagref, Aubière (France).

I also would like to thank the numerous contributors and organizers of specific workshops and sessions, for their efforts.

Finally, I would like to thank Philippe Geril, the EUROSIS coordinator, who managed the major part of the conference organization and also Alain Piel, the Sciences and Technologies Faculty Director who has graciously accepted for the conference sessions to be held in his Faculty.

Last but not least, we would like to wish you a lot of interesting scientific discussions from these contributions and also a pleasant stay in Le Havre, which has recently been labelled by Unesco World Heritage as a site of historic importance for its complex past, both as a harbour logistics platform and a complex estuarian environment.

Cyrille Bertelle
ESM 2008 General Conference Chair
CONTENTS

Preface ..........................................................................................................................XI
Scientific Programme .................................................................................................1
Author Listing ...............................................................................................................589

SIMULATION METHODOLOGY

VERIFICATION, VALIDATION AND OPTIMIZATION

Requirements related to the Validity of a Simulation
Vincent Albert, Alexandre Nketsa, Mario Paludetto and Marc Courvoisier .......... 7

A Graph-Based Approach to Verification and Validation of Simulation
Models and Applications
Zhongshi Wang and Axel Lehmann ...........................................................................14

Using Integer Linear Programming for Discrete Problem Optimization
Jaroslav Sklenar, Valerie Cutajar and Milan Ceska .................................................19

MODEL REFINEMENT

Emergence as Metaknowledge: Refining Simulation Models through
Emergence Reification
Daniel David and Rémy Courdier ...........................................................................25

Analysis and Simulation of the Leg of an Hexapod Robot for Remote
Exploration
J. Torres, G. Romero, J. Gomez-Elvira and J. Maroto .............................................28

Localisation of Joint Rotation Centres for 3D Human Motion Simulation
Martin Pruzinec, Hildegard Köhler and Annika Wörner .......................................35

Managing the Resolution of Simulation Models
Gábor Lencse and László Muka ................................................................................38

MODELLING METHODS IN INDUSTRY

Modelling of compaction and residence times in continuous cooking
application
Miika Uusitalo and Timo Ahvenlampi .....................................................................45

Influence of the Geometry of the Workpiece on Contact Pressures in
Frictionless Indentation Processes
Marta M. Marín, Ana M. Camacho, Eva Rubio and Miguel A. Sebastián .......... 51
CONTENTS

An Activity- and Actor-Oriented Approach for the Project Engineering of Complex, Weakly Structured Projects
Sven Tackenberg, Sönke Duckwitz, Selvinaz Karahancer, Wolfgang Kabuß, Bernhard Kausch and Christopher M. Schlick .......................................................... 56

Actor Oriented Person centered Simulation of Product Development Projects
Sönke Duckwitz, Torsten Licht, Peter Schmitz and Christopher M. Schlick...... 66

The Effect of the Model Composition to the Structural Properties of Process Models
Adrien Leitold and Miklós Gerzson .................................................................. 74

Computer Simulation of Cross-Sectional variation for worsted Wool Type Yarn
Ionut Neagu, Pascal Bruniaux and Dorin Avram ................................................ 82

BEHAVIOUR METHODOLOGY

On the Use of Association Rule and Clustering in Student Progression
Kifaya S.Qaddoum ................................................................................................. 89

Executing UML 2 Diagrams in ActiveCharts - A formal Semantics for the Combination of Behavior Specifications in the UML 2
Jens Kohlmeyer .................................................................................................... 94

SIMULATION TOOLS

SOFTWARE TOOLS

Dispatching Independent Railway Simulations with DIRS: A simple RSH Based Tool
Damien Feillée, David Hill and Gilles Dessagne ................................................. 107

Simulation for an Operator’s Decisions in SCADA
Andrzej Urbaniak ................................................................................................ 112

From Design with SYSML to VHDL-AMS Simulation
Jean Verries, Mario Paludetto and Abd-El-Kader Sahraoui ............................... 115

Prediction Tools for Student Learning Assessment in Professional Schools
Paulo Almeida, Paulo Novais and José Neves.................................................. 121
CONTENTS

EVALUATION AND CLASSIFICATION TOOLS

A Multiscale Simulation Environment for Performance Evaluation of high reliable heterogeneous Communication Networks
Andreas Lewandowski, Ralf Burda, Sebastian Subik and Christian Wietfeld ... 131

Service Level Agreement Specification, Compliance Prediction and Monitoring with Performance Trees
N.J. Dingle, W.J. Knottenbelt and L. Wang ......................................................... 137

Simulation and Modelling of RAID 0 System Performance
F. Wan, N.J. Dingle, W.J. Knottenbelt and A.S. Lebrecht ................................. 145

A New Generation of Nuclear Power Plant Simulation Applications
Mireya Rodríguez, José A. Cortés, Manuel Díaz, Daniel Garrido and Enrique Soler ................................................................. 150

SIMULATION AND AI

INDUSTRIAL APPLICATIONS

Diagnosis of extraction screen plugging in continuous cooking application
Timo Ahvenlampi and Miika Uusitalo.............................................................. 161

Interactive Simulation System for Testing Washing Machine Designs
Carla Martin-Villalba, Alfonso Urquia, Sebastian Dormido and Félix Martínez 168

BEHAVIOURAL AI

Simulating and Monitoring Ambient Assisted Living
Davide Carneiro, Ricardo Costa, Paulo Novais, João Neves, José Machado and José Neves................................................................. 175

A Simulation-Based Model of Abduction
Gildas Morvan, Daniel Dupont and Philippe Kubiak ........................................ 183

NEURAL NETWORKS AND AI TECHNIQUES

Grammatical Evolution for Development of Neural Networks with Real Valued Weights using Cellular Encoding
Jan Drchal and Miroslav Snorek ....................................................................... 191
CONTENTS

Naive Baysian and K-Nearest Neighbour to Categorize Arabic Text
Wa’el Musa Hadi, Fadi Thabtah, Samer ALHawari and Jafar Ababueh ...................... 196

Energy Efficient PSO-based Algorithm for Optimizing Autonomous Wireless Sensor Network
Mohaned Al-Obaidy and Aladdin Ayesh ......................................................................... 201

ARTIFICIAL SOCIETY

TRAFFIC BEHAVIOUR

A high Performance Framework for Agent Based Pedestrian Dynamics on GPU Hardware
Paul Richmond and Daniela M. Romano ......................................................................... 213

Modeling Pedestrians Movement regarding Road Crossing Compliance
Carlos O. Pretto; Helena B. B. Cybis and Paula Ariotti ............................................. 221

An agent-based simulation framework for modelling travel behaviour in Flanders
Katrien Ramaekers, Tom Bellemans, Davy Janssens and Geert Wets ......................... 226

A General Framework for Routing Problems in Real Traffic Conditions
Michelle Chabrol, Michel Gourgand and Patrice Leclaire ........................................... 230

A Tool for Car Traffic Management
Alfredo Cutolo, Raffaella Frattaruolo and Luigi Rarità ................................................ 238

COLLECTIVE BEHAVIOUR

Ant Colony Optimization with Parallel Subsolutions Heuristic
Oleg Kovarik and Miroslav Skrbek ................................................................................ 247

Simulation of Immune System Response to Bacterial Challenge
Nuno Fachada, Vitor V. Lopes and Agostinho Rcsa ...................................................... 252

Urban Cultural Dynamics Modelling using Swarm Intelligence and Geographical Information Systems
Rawan Ghnemat, Françoise Lucchini and Cyrille Bertelle ........................................... 258

Managing Spatial Self-Organization via Collective Behaviors
Rawan Ghnemat, Cyrille Bertelle and Gerard H.E. Duchamp ...................................... 264
CONTENTS

Multiple Virtual Command Centres for Controlling Physically Distant Set of Robots: Next Generation of Telepresence
G. Al-Hudhud .................................................................................................................. 267

AGENT BASED SIMULATION

AGENT BEHAVIOUR

Modeling the Emotional State of an Agent through Fuzzy Logic With Reference to the Geneva Emotion Wheel
William F. Biewitt and Aladdin Ayesh ................................................................. 279

Learning Agent Behavior for Agent Based Simulation using Genetic Algorithms
Benjamin Wolters and Timo Steffens ................................................................. 284

AGENT BASED MANAGEMENT

Quality of Service in Healthcare Units
José Machado, António Abélha, Paulo Novais, João Neves and José Neves . 291

Enterprise Knowlegde Management supported by Agent Based Modelling and Simulation
Marco Remondino and Marco Pironti ................................................................. 299

AGENT BASED MODELS

A based words and automata model for infinitely repeated games
M. Arfi, B. Ould M Lemine and C. Selmi .......................................................... 307

Definition of Operating Procedures in Generic Agent Based Simulation Model of Transportation Logistic Terminals
Norbert Adamko and Valent Klima ...................................................................... 312

A New Probabilistic Combat Model of Direct Fire with Random Fire Distribution
B. Nguyen and M. Rempel .................................................................................... 317

PARALLEL PROCESSING

Hardware Acceleration for Computational Intelligence - THSOM Neural Network on x86 hardware
Rudolf Marek and Miroslav Skrbek ..................................................................... 327
CONTENTS

Simulation Modeling of a Cluster to Cluster Grid
Lazar Kirchev and Vasil Georgiev ................................................................. 332

Multilayered and heterogeneous modelling and simulation of natural complex systems
Emilie Broutin, Paul Bisgambiglia and Jean-François Santucci ...................... 338

Analysis of Temporal Dependencies of Perceptions and Influences for the Distributed Execution of Agent-Oriented Simulations
Nicolas Sebastien, Remy Coudrier, Didier Hoareau and Marc-Philippe Huget ................................................................. 343

On the Development of Object Oriented Parallel Penta-Diagonal Preconditioned Krylov Subspace (K-S) Solvers
Bipin Kumar, Martin Crane and Yan Deallure .................................................. 348

SIMULATION IN EDUCATION

Model-Based Learning in Electric Drive Design
Valery Vodovozov and Zoja Raud ................................................................. 355

Visualizing the Evolution of Warehouses
Pietro Colombo and Alberto Coen-Porisini ..................................................... 360

ANALYTICAL AND NUMERICAL MODELLING TECHNIQUES

Analysis of waiting times in cellular mobile systems with retrials and guard channels
M.J. Lopez-Herrero ......................................................................................... 369

A Bandwidth Management Framework for Wireless Mobile Ad Hoc Networks
Benny M Nyambo, Gerrit K. Janssens and Wim Lamotte ................................ 373

Modeling and Simulation of Advanced Floating Body z-Ram Memory Cells
Viktor Sverdlov and Siegfried Selberherr ........................................................ 380

Distribution of the successful and blocked events in the M/G/ 1 retrial queue
J. Amador ........................................................................................................ 385

COMPLEX SYSTEMS SIMULATION

A deterministic Annealing Algorithm for the Pre-and End Haulage of Intermodal Container Terminals
An Caris and Gerrit K. Janssens ........................................................................ 391
CONTENTS

Modelling and Simulation of the Pharmacy for the University Hospital Center of Oran (Algeria)
Khaled Belkadi and Alain Tanguy ................................................................. 399

Cooperating Modelling Methods for Performance Evaluation of Interconnected Infocommunication and Business Process Systems
László Muka and Gábor Lencse ................................................................. 404

Modelling of Gas Turbine Based Plants during Power Changes
Henryk Kapron and Michał Wydra ............................................................ 412

SIMULATION IN BIOLOGY

Thermoregulation in P Terraenovae Aggregations: An Agent Based Approach
Gildas Morvan, Daniel Jolly and Damien Charabidze .................................. 417

Gene regulatory networks with multiplexes
Gilles Bernot, Jean-Paul Comet and Zohra Khalis ........................................ 423

FLUID FLOW SIMULATION

Water Hammer in Nuclear Installations Case Study in Feed WaterSystem
Ilie Prisecaru, Daniel Dupleac and Nîţă Iulian Pavel .................................. 433

Mathematical Models for Hydraulic Calculation and Optimization of Communal Waternets
Lucyna Bogdan and Jan Studzinski ........................................................... 438

PETRI NETS SIMULATION

STOCHASTIC PETRI NETS

From BCMP Queueing Networks to generalized Stochastic Petri Nets:
An Algorithm and an Equivalence Definition
Simonetta Balsamo and Andrea Marin ....................................................... 447

Timed Stochastic Petri Net Models Identification for Simulation and Reliability Analysis
Souleiman Ould El Medhi, Edouard Leclercq and Dimitri Lefebvre ............... 456
## CONTENTS

### PROJECT MANAGEMENT

**Object Oriented Petri Nets Usage in Project Portfolio Modeling Domain**  
Šárka Květoňová and Vladimír Janousek .......................................................... 467

**Timed Coloured Petri Nets use in Project Management Formalization Domain**  
Šárka Květoňová and Dušan Kolář ................................................................. 473

**A Service Based Enactment Engine for Dynamically Reconfigurable Workflows**  
Franco Cicirelli, Angelo Furfaro and Libero Nigro ........................................ 479

### PETRI NETS MODELS

**Algorithms for Computing Coverability Graphs for Continuous Petri Nets**  
Petr Novosad and Milan Česka ........................................................................ 489

**Continuous Approximation of PEPA Models and Petri Nets**  
Vashti Galpin .................................................................................................... 492

**A performance driven modeling approach for SOA based application**  
Mauro Iacono and Stefano Marrone .............................................................. 500

### EMERGENCY RISK MANAGEMENT

### LARGE SCALE SYSTEMS

**Revealing the impact of information provision on disaster relief operations by means of agent-based modeling and simulation**  
Gerald Lichtenegger and Siegfried Voessner .................................................... 511

**DT-MANETs for Rescue Information Service**  
Julien Franzolini, Frédéric Guinand and Damien Olivier ................................. 516

**The Simulation of the Procedure for Multiattribute Choice of the Structure of the Power Plants in the Power System**  
Eugeniusz M. Sroczan ...................................................................................... 521

**Modeling of the Vulnerability related to the Dynamic Road Traffic**  
Cyrille Bertelle, Antoine Dutot, Michel Nabaa, Damien Olivier and Pascal Mallet ................................................................................................................ 524
CONTENTS

SPECIFIC MODELS

System Dynamics versus Cellular Automata in Modelling Panic Situations
Damienne Provitolo and Eric Daudé ................................................................. 535

An Architecture for Individual Behaviour Modeling in Emergency Situations
Julien Valentin, Michael Bergeret, Eric Guardères and Wilfrid Lefer ............... 541

Fitting of Segmented Gaussian Plume Model Predictions on Measured Data
Petr Pecha and Radek Hofman ........................................................................ 544

SIMULATION, MODELLING AND DATAMANAGEMENT IN REAL-TIME SYSTEMS

RTDS: A component and Aspect-based Real-Time Database System Simulator
Leila Baccouche and Sami Laimam ................................................................. 551

A Fault Tolerant Replication Control Protocol for Distributed Real-Time Database Systems
Anis Haj Said, Bruno Sadeg, Laurent Amanton and Bechir Ayeb .................... 556

Data Model and Real Time in Spatial Applications: What Data and When?
Michel Mainguenaud .................................................................................. 561

From a Master Server Architecture to a Feedback Control Architecture
Bechir Alaya, Claude Duvallet and Bruno Sadeg ........................................... 566

TUTORIAL

Design and Analysis of Computer Experiments with SimExplorer
Florent Chuffart, Thierry Faure, Nicolas Dumoulin, Romain Reuillon and
Guillaume Deffuant .................................................................................. 575

LATE PAPER

The GeoCells Project
Patrice Langlois, Dominique Goyat and Bernard Elissalde ............................ 581
SCIENTIFIC PROGRAMME
SIMULATION METHODOLOGY
VERIFICATION
VALIDATION
AND
OPTIMIZATION
REQUIREMENTS RELATED TO THE VALIDITY OF A SIMULATION

Vincent Albert, Alexandre Nketsa, Mario Paludetto, 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|marc}@laas.fr

KEYWORDS
Validity, Abstraction, Matching, Typing

ABSTRACT
The paper aims to define and formalize a part of requirements related to the validity assessment of a simulation in the context of real-time systems development. Validity 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. We have refined the concept of validity along four domains – architecture, data, computation and time. Simulation users express their requirements about simulation validity over those domains and simulation developers express the simulation properties over the same domains. We have to formalize domains description, and proposed matching criteria to assess if the simulation properties allow reaching the intended purpose of the user of this simulation. This paper focuses on two of these domains.

INTRODUCTION

Why use simulation?
The most recognized advantages of using Modeling and Simulation (M&S) for complex embedded systems development are:

- The capability to predict and validate the behaviour of a system sooner in the development cycle,
- The capability to check systems to systems interactions or before system integration on the physical (final) target mean.

As a consequence, simulation allows to:

- Improve system concept robustness,
- Anticipate systems integration,
- Speed-up design iterations and then allow better choices and trade-off for overall system optimization,
- Reduce the number of physical tests.

When use simulation?
As others methods like reviews, analysis and formal verification, simulation is used in system engineering, to verify and validate a specification, a design or the implementation of a system, namely the system of interest. Verification addresses whether the system, its elements, its interfaces, and incremental work products satisfy their requirements. Validation confirms that the system, as built (or as it will be built), will satisfy the user’s needs. Verification ensures the conformance to those requirements, and validation ensures the requirements and the system implementation provide the right solution to the customer’s problem. (EIA-731 1998). In other words, verification ensures that “you built it right” while validation ensures that “you built the right thing.”

What is simulation validity assessment?
In system engineering, validation is the activity that consists to make sure, essentially by tests means, that a product is conforming to its specification or the product is satisfying the user requirements. The M&S tradition associates validation as the activity to make sure that a model or a simulation has behaviour similar to the system of interest. Validation deals with the examination of fidelity (Pace 1999), suitability (Brade 2004) or validity (REVVA 2004). In fact in M&S community validation is part of a wider definition of simulation assessment, namely Verification, Validation and Accreditation (VV&A) (DMSO 2000). Our scope is limited in this paper to the validation part.

Why simulation validity assessment is important?
It is important to note that the validity of a simulation means relies on the compliance of these characteristics with the intended use expressed by the user. Indeed, the purpose of a simulation is to represent a real system in order to draw conclusions about the real system by experimentation with a model. Thus, the direct correlation between the model, an intended purpose of model use, and a clearly identified real system are among the key characteristics of simulation. The term “simulation” implies a claim to represent the behaviour of a real system as it is or as it could be.

In that context, simulation results must only be used, if they are sufficiently credible with respect to the impact of their use. If the influence of the model, simulation results, or observed model behavior is high, wrong or unsuitable simulation results are not compensated by conventional information, and most probably lead to wrong decisions with undesired consequences.

M&S: AN ABSTRACTION OF THE REALITY
A major thing which differentiates a simulation from the real world is the consequence of M&S hypotheses. There are, broadly speaking, two kinds of hypotheses. Those dedicated to simplify a system too complex in order to adapt it to a given objective and those which are due to a lack of knowledge of the system at a given development phase of that system.
Whatever the origin of these hypotheses, the underlying limitations must be identified and documented in order to describe the Simulation Domain of Use (SDU). To solve this description problem of a SDU, we had, in (Albert et al. 2007) dedicated our investigations to the types of M&S hypotheses and their impacts on the simulation properties.

Such properties description would be in the tradition of most of M&S strategic plans:

- To have the right systems models each time they are needed.
- To ensure all the models for a same system are consistent whatever the stage, the level and the discipline.
- To allow the retro feeding of the various models with properties valued later in the development (models developed in the early stages contain hypotheses which have later to be discharged when the design of the system is complete).

Recall that a simulation is valid (or not) according to a simulation user intended purpose. Then, if simulation developers have to provide a rigorous documentation about the domain of use of their products, on the other hand simulation users must provide rigorous requirements about the expected degree of validity of the needed simulation products. Again such requirements description would be in the tradition of most of M&S strategic plans. Indeed, the most “realistic” simulation product is always preferred for validation without taking care of its cost and availability date. A rigorous requirements description would improve the simulation use and development by designing the sufficient and necessary simulation product.

Then, M&S validity assessment underlines two main issues:

- Formally describe what a model and the simulation constructed around this model represent, namely the Simulation Domain of Use (SDU).
- Formally describe simulation user intended purpose as expected degree of validity, namely the Simulation Objectives of Use (SOU).

Having, on one hand simulation properties and limitations and on the other hand simulation objectives of use, one can apply matching criteria to assess the M&S validity according to the objectives of use.

**SIMULATION VALIDITY ASSESSMENT PROCESS**

We have focused our validity assessment process on the valid interaction between the SDU and the SOU. The validation establishes with which degree the simulation properties match with the intended purpose. It is worth noting that this validation activity can be made before or after the simulation development, in the case of a simulation product reuse.

The figure 1 below illustrates the process. Given its intended purpose, which concretely relies on system of interest documents (requirements, specifications, V&V plan), the simulation user build its experimental frame (EF) (Zeigler et al. 2000). The experimental frame control, configure and stimulate the simulation in terms of input trajectories, scenarios, parameters, observations... Then the experimental frame leads to the admissible abstractions to reach the intended purpose. For example if the intended purpose is to validate one independent function of the system of interest and only that function, you can make abstraction about the rest of your system domain. Then admissible abstractions lead to the SOU (required simulation properties and expected result quality). This is illustrated on the left-hand side of the figure. The right-hand side of the figure illustrates the simulation developer concern. Usually given the modelled system requirements and specifications, and by tracing given abstractions and simplifications, a formal description of the SDU is given.

![Diagram](image)

**Figure 1. Simulation Validity Assessment Process**

1 The EF diagram on figure 1 is taken from (SISO 2007)
Then through a set of criteria one can show that properties related to SOU matches with SDU. It is worth noting that SDU and SOU are described with same kind of properties along four domains (architecture, data, computation and time) in order to allow the matching assessment. SOU/SDU description format will play the role of simulation conceptual model (Harmon 2006; Sargent 1987; Pace 1999) for embedded systems. Being the first product of the design process that describes the actual capabilities that the simulation has or will have (or must have), a simulation conceptual model is frequently described as the bridge between the developer and the user (DMSO 2000).

**METHODS**

Consider the figure 2 below (SISO 2007). The experimental frame specifies the conditions under which the system is observed or experimented. We can see it as a system that interacts with the SEM (Simulation Executable Model) in order to obtain the required data under specific conditions.

![Diagram of a simulation within a specific experimental frame](image)

**Figure 2.** Diagram of a simulation within a specific experimental frame

SDU is an assessment-oriented documentation of the SEM. SOU is an assessment-oriented documentation of the EF. Recall that our approach of simulation validity assessment is to assess the valid interaction between the SDU and the SOU. The basic idea is to enrich the SEM interfaces (inputs and outputs), in order to define through its interfaces and only through its interfaces its features and known limitations. Such specification is done by the simulation developer. On the other hand the idea is to enrich the EF interfaces (stimuli and observers), in order to define the required capabilities of the simulation. SOU specification is done by the simulation user. SEM and EF extended interfaces, must covers the four domains identified above to assess the validity. Component Based Software Development (CBSD) community matches with our issue.

Component Based Software Development (CBSD) composes a software system which satisfies given requirements, using available components. When applying CBSD to large-scale applications, we often are faced with the difficulties that granularity and functional boundaries are different between requirements and components. These differences make component selection and software system composition difficult. Then, the community focused its works on formal approaches relieving the above difficulties in order to compose component-based systems effectively.

A software component for which Szyderski’s definition is now generally accepted, at least in the industry is: a unit of composition with contractually specified interfaces and explicit context dependencies only. Through and only through its interfaces it is connected to its environment. Component-based formal approach aims at revealing the properties of a component within its interfaces. It consists in enriching such interfaces in order to provide a syntactic and semantic signature of the component.

Signatures play the same role in CBSD as type signatures in computer programming. A data type signature defines the inputs and outputs for a function or method. This kind of type signature includes at least the function name and the number of its parameters. In some programming languages, it may also specify the function's return type or the types of its parameters. Data type signature only covers the static structure of a component (this is what we call the syntactic signature). Richer interfaces for component, namely behavioural type signature, have been defined in order to cover behavioral properties (Talcott 1996; de Alfaro and Henzinger 2001; Lee and Xiong 2002; Arabab 2005), namely the semantic signature. The semantic signature captures the dynamic interaction of components (e.g. relative order in which a model expects their functions to be called). We use successively such formal methods in order to specify our components (SOU and SDU) and cover the four domains of analysis. Then, data and behavioural type signature can be used to check the compatibility of components.

Then assume that SOU and SDU are components as defined by Szyderski. A component receives tokens from input ports and reacts to these tokens by producing tokens on the output ports. Components are composed by connecting ports with connectors (figure 3). The use of connectors to mediate communication implies that components interact only with the connectors that they are connected to and not directly with other components. Then, we define matching criteria as signature conformity and finally, we assess matching criteria by signature conformity analysis.

![Component description and composition](image)

**Figure 3.** Component description and composition

**SIMULATION CONCEPTUAL MODEL FOR REAL-TIME SYSTEMS**

A domain is an aspect of a model which can logically be analyzed independently from other aspects (Jantsch et al. 1999). Model abstraction is a method for reducing the complexity of a simulation model while maintaining the validity of the simulation results with respect to the question that the simulation is being used to address (Frantz 1995). There are many domains along which a model can be abstracted. We have built our simulation conceptual model along four important domains mainly found within artificial intelligence community (Iwasaki 1990; Weld 1992):
architecture, data, computation and time. Briefly, the architecture domain focuses on the domain boundaries of the simulation and its topology. Data domain focuses on the representation level of simulation interfaces. Computation domain focuses on the way the results are computed. It is concerned with the relationship of input and output values. Time domain aims to define the required temporal granularity so as to capture the events of interests of the application. We focus in this paper on architecture and data domains formalization.

**Architecture validity**

Architecture validity is depicted in two concepts: scope and topology.

**Scope**

We retain the following definition from the Artificial Intelligence community (Weld 1992). Others definitions can be found in (DMSO 2000).

*Scope denotes the range of the phenomena that the model describes. A model has a greater scope than another if it describes strictly more of the world. Changing a model’s scope is equivalent to recasting the boundary between the system (described by the model) and the environment. Thus, implicit in the choice of scope is the selection of exogenous parameters and the determination of their driving values.*

The definition given by Weld introduces an important concept. That is when changing the model’s scope, we also change the system domain that is modelled. By changing the system domain, we are recasting the boundaries between the system and the environment. In other words we are defining new required inputs and new expected outputs in order to carry out the experimentation.

Consider a system A (figure 4a) that has a set of endogenous parameters (cross) and a set of exogenous parameters (circle). When reducing the model’s scope we will obtain the system B (figure 4b) with a new (smaller) set of endogenous parameters and a new (bigger) set of exogenous parameters.

![Figure 4. Models with differing scope](image)

New exogenous parameters (black circle) become parameters controlled and/or monitored by the simulation user in order to carry out the experimentation. Thus we can argue that the model’s scope is fully defined by the set of exogenous parameters.

Then, let:
- $I_{SOU}$ be the set of expected input ports (observers),
- $O_{SOU}$ the set of required output ports (stimuli),
- $I_{SDU}$ be the set of provided input ports,
- $O_{SDU}$ the set of provided output ports.

We assume that the sets $I_{SOU}$, $O_{SOU}$, $I_{SDU}$ and $O_{SDU}$ provide only names.

Then one can define the required scope to reach a intended purpose $Scope_{SOU}$ as the set of exogenous parameters, namely, the set of expected input ports to monitor the simulation and the set of required output ports to control the application

$$Scope_{SOU} = I_{SOU} \cup O_{SOU}$$

We assume that the scope of the SDU is defined by the set of input and output ports provided by the simulation product

$$Scope_{SDU} = I_{SDU} \cup O_{SDU}$$

Assume that $Scope_{SOU}$ and $Scope_{SDU}$ are a first step for signature. Then the matching criterion for scope domain is defined by

$$Scope_{SOU} \subseteq Scope_{SDU}$$

**Topology**

One can describe the simulation topology with causality interfaces (Lee et al. 2005). Developed for composition analysis, causality interfaces is a special family of behavioral interfaces that capture the causality properties of actors, which reflect the dependency of particular outputs having on particular inputs. Nevertheless we use this formalism only for static properties. Dynamic properties of causality will be used by another domain, namely computation.

The general definition given by (Lee et al. 2005) is as follows:

Assume an ordered set $R$ with elements called dependencies with $R = \{true, false\}$.

A causality interface for a component $a$ with input ports $I$ and output ports $O$ is a function

$$\delta_a : I_a \times O_a \rightarrow R$$

A causality interface for a connector $c$ that links output ports $O$ of a component $a$ to input port $I$ of a component $b$ is a function

$$\delta_c : O_a \times I_b \rightarrow R$$

Consider an SDU with a scope defined by $Scope_{SDU}$ and a SOU with a scope defined by $Scope_{SOU}$. Assume a specific topology defined by

$$\delta_1 : O_{SOU} \times I_{SDU} \rightarrow R$$

$$\delta_2 : O_{SDU} \times I_{SOU} \rightarrow R$$

We say that such a required topology is recoverable if and only if

$$\forall p_o \in O_{SOU} \exists p_i \in I_{SOU} : \delta_1 = true$$

$$\forall p_i \in I_{SDU} \exists p_o \in O_{SDU} : \delta_2 = true$$

10
Interpretation (figure 5):

$\delta_t(p_o, p_i)$ denotes the dependency that port $p_o \in O_{SOU}$ has on port $p_i \in I_{SOU}$.

$\delta_s(p_o, p_i)$ denotes the dependency that port $p_o \in O_{SDU}$ has on port $p_i \in I_{SDU}$.

![Figure 5. Simulation topology](image)

**Data Validity**

A simulation handles many different types of data, at different levels of representations. It is necessary when using a simulation to specify the levels of representation of the required input ports and expected output ports and the abstract set of possible values that a port may assume. The difficulty here remains on the heterogeneity of the levels of representation. Such levels can refer to the unit of measure of the data, its precision, the device (e.g. specific Man Machine Interface) used to handle it.

A simulation consists of a set of inter-connected components with well-defined ports and an explicit experimental frame which defines the borders of the system under test. Data domain focuses on connection and communication between two ports. We have assumed that components send and receive messages through ports via connectors, and data object are encapsulated in tokens. Output ports send out tokens and input ports receive tokens. Data domain aims at constraining the type of object from being used with a port, namely the datatype. Then matching criteria for data domain consists in assessing that inter-connected ports have compatible types, e.g. if a component expects to receive data encoded as integers, but another component sends it a string, then the first component may not be able to function correctly.

As we said previously datatypes have many concerns. At an abstract level it specifies the concept that is transported by the data (e.g. altitude, speed, energy, fuel quantity, power, display). The amount of information it contains is completely unspecified. It can be seen as polymorphic data in OOP languages where the exact type of the data is known when it is instantiated for execution. Abstract level also specifies the way the simulation user has access to the data (e.g. specific MMI, physical signal…).

At a mathematical level, datatypes are based on number systems (i.e. natural, complex, real, integer…) it consists on giving the following information about the data: unit, range, spacing and precision:

- the range of the data to be stimulated or observed refers to the overall coverage of the data and gives the acceptable limits of the representation
- the spacing of the data refers to the distance between successive samples in the representation
- the precision of the data refers to the distance between the physical value and the value of higher representation level.

Such considerations are very important in simulation validity assessment. Let us take the simple example of a flight management system giving the position of an aircraft according to a given waypoint where the true value is D=100m. Consider that the model gives a value of 98m. If the model resolution is infinitesimally high (resolution error is 0), then one can say that the model is inaccurate (computation domain concern). Let us now suppose that the model gives the value of D with a resolution error equal to 2m ([98,102]), then the model is accurate as the true value is in the interval. Then we argue that this domain is an important validity aspect in order to indisputably state on the sources of uncertainty (computation validity or data validity).

Then the data is encoded in a digital referential (a finite suite of bits). We call it the computer level. This level includes types normally associated with a programming language such as integer, boolean and char.

We assume a set of sorts D containing data concepts. We assume that set D provides only names, e.g.

\[ D = \{ \text{altitude, speed, energy, fuel quantity, power, display} \} \]

Then we define a function $\text{fct}$ which associates to each of the I/O ports an element of D

\[ \text{fct} : I_{SOU} \cup O_{SOU} \cup I_{SDU} \cup O_{SDU} \to D \]

We assume a set of sorts Rg containing ranges. For example a tension is limited by [-10V,10V] or a valve surface is limited by [0,S\text{max}].

We define a function $\Omega$ which associates to each of the I/O ports an element of Rg

\[ \Omega : I_{SOU} \cup O_{SOU} \cup I_{SDU} \cup O_{SDU} \to Rg \]

Interpretation: $\Omega (p_i)$ denotes the range (domain of value) associated with the port $p_i$.

We assume a set of sorts Sp containing spacing. For example a tension that is limited by [-10V, 10V] can with a spacing of 5V can only take the values \{-10V,-5V,0V,5V,10V\}.

We define a function $\sigma$ which associates to each of the I/O ports an element of Sp

\[ \sigma : I_{SOU} \cup O_{SOU} \cup I_{SDU} \cup O_{SDU} \to Sp \]

Interpretation: $\sigma (p_i)$ denotes the spacing associated with the port $p_i$.

We assume a set of sorts P containing precisions. For example assume an altitude represented with flight phases ECAM system logic data. Flight phases 5 and 6 respectively correspond to altitude ranges [0;1500ft] and [1500ft;+\infty]. In
that specific case, we have a spacing equal to 0 and a precision equal to 1500ft for the flight phase 5 and + ∞ for flight phase 6.

Then we define a function ρ which associates to each of the I/O ports an element of P

\[ \rho : ISO \cup OSO \cup ISD \cup OSD \rightarrow P \]

Interpretation: ρ(p) denotes the precision associated with the port p.

Data domain matching criteria consists in assessing that inter-connected ports have compatible types. The compatibility rule between SOU and SDU imposes a sort constraint across every connection from an output port to an input port. Assume a set δ containing the causality interfaces for a given topology, then

\[ \forall a \in \delta(p_0, p) \in OSO \times ISO \cup OSO \times ISO / fct(p) = fct(p) \]
\[ \forall a \in \delta(p_0, p) \in OSO \times ISO \cup OSO \times ISO / \delta(p) \geq \delta(p) \]
\[ \forall a \in \delta(p_0, p) \in OSO \times ISO \cup OSO \times ISO / \sigma(p) \geq \sigma(p) \]
\[ \forall a \in \delta(p_0, p) \in OSO \times ISO \cup OSO \times ISO / \rho(p) \geq \rho(p) \]

For computer level matching criteria we use a type lattice (Xiong 2002) as illustrated in figure 6. The type lattice establishes a partial ordering among types. The upward lines denote ≥, a loss of information. For example a 32 bits integer can be converted to a double without losing information while the reverse is not true (int ≤ double).

Then we assume a set of sorts T containing “computer-based” datatypes, e.g.

\[ T = \{ \text{int, complex, long, double, string} \} \]

We define a function τ which associates to each of the I/O ports an element of Sp

\[ \tau : ISO \cup OSO \cup ISD \cup OSD \rightarrow T \]

The compatibility rule between SOU and SDU requires that the sort of an output port, τ(p0), be the same as the sort of the inter-connected input port, τ(p), or at a lower level than τ(p).

\[ \forall a \in \delta(p_0, p) \in OSO \times ISO \cup OSO \times ISO / \tau(p) \geq \tau(p) \]

**APPLICATION CASE**

We had experiment the process and our method on an application case. The system of interest was an Air-Traffic Control (ATC) system. ATC system is composed by a set of applications in order to provide different services to the pilots to communicate with the ground. Then it is interfaced with a few numbers of others systems. As an input we had in a textual format, the ATC specification the V&V scenarios and the architecture of the simulation supposed to allow realizing the tests. The first part of the experimentation has consisted on specifying the needed architectural domain in order to realize a part of the V&V scenarios. For each script tests we identified the required output ports and expected input ports and we specified their datatypes. The results were relevant. Indeed it appears that a few script tests could not be realized on that specific simulation platform. That was mainly due to missing functionalities in the simulation models. The studies also reveal that some of the interfaced systems modelled and implemented in the simulation were not used during the V&V scenarios. One could status that the simulation was too complex for that intended purpose.

**CONCLUSION**

This paper has addressed the problem of defining the requirements for assessing the validity of a simulation with regards to the simulation intended purpose. We have proposed a general approach based on formal documentation matching to assess such validity. This formal documentation is depicted into four domains – architecture, data, computation and time. We have presented architecture and data domains. Computation and time domain has been treated in the same analytical mind. Nevertheless we have, to date, defined only static properties. Further work would consist on making a projection of our domains on dynamic properties. Dynamic properties would give a behavioural description of the computations, i.e. the application functions, the models of computation and communications/synchronisations between simulation components.

Specifically, within the dynamic part we capture:

- SDU point of view - assumptions about the input behavior and guarantees about the output behavior of the simulation.
- SOU point of view – output behavior that will be injected in the simulation and expected input behaviour.

As explained in the paper we whish using behavioural type signature to check the dynamic compatibility of components.
REFERENCES

SISO, 2007. “Guidance for a Generic Methodology (GM) for Verification and Validation (V&V) and Acceptance of Models, Simulations, and Data”. Technical notes
A GRAPH-BASED APPROACH TO VERIFICATION AND VALIDATION OF SIMULATION MODELS AND APPLICATIONS

Zhongshi Wang$^1$ and Axel Lehmann$^2$
(115,314),(417,328)

1 Institut für Technik Intelligenter Systeme (ITIS)
2 Institut für Technische Informatik
Universität der Bundeswehr München
D-85577 Neubiberg, Germany
{zhongshi.wang, axel.lehmann}@unibw.de

KEYWORDS
Verification and Validation (V&V), Quality Assessment, Graph Theory, Model Defect Classification.

ABSTRACT

Model defects, if not detected and removed in time, propagate through the development life cycle of modelling and simulation (M&S). The later they are detected, the more problems they cause, and the more expensive and time-consuming is their correction. A variety of verification and validation (V&V) processes has been proposed for the purpose of detecting as many different defects as possible in the early M&S development phases. Unlike those approaches, this work focuses on tracing the dependencies of associated defects and proposes a generic approach to establishing dependency chains in terms of a directed graph. Based on the dependency chains, all associated defects can be tracked and identified. Applied in combination with other existing V&V processes, this approach provides an additional opportunity to detect and correct all the related defects of M&S work products consequently.

INTRODUCTION

Development of modelling and simulation (M&S) applications requires intensive investment of human efforts. In human-based activities, defects, which lead to quality deficiencies in models and simulation results, are inevitable and need to be systematically detected, tracked, and removed (Biffi and Halling 2003). Verification and validation (V&V) focuses on assessing the accuracy quality characteristic of an M&S application with respect to its objectives (Balci 2004, Shannon 1975), and is intended to ensure that only correct and suitable models and simulation results are used.

Identifying and removing model defects as early as possible in the life cycle of a simulation study is a crucial issue of V&V (Balci 1998), since an undetected defect influences the successive documents and causes more defects in the next development phase (Beizer 1990, Megen and Meyerhoff 1995). The later defects are detected, the further they have propagated through the M&S life cycle, and the more resources are required for their correction.

The forms of defect propagation are twofold: (1) a defect is adopted in the same or similar form by the subsequent defects, e.g., an error value can be transferred to other documents, and (2) an incorrect or inaccurate assumption of an early phase is further interpreted, from it new (certainly in this case also erroneous) conclusions are deduced, and then applied to develop other model elements. Detecting and tracing suchlike associated defects is extremely complicated. Nance (Nance 1994) points out in a study that some vital defects, if not detected in time, may not even be trackable in later phases.

To address this problem, many different V&V approaches have been introduced, which typically focus on (1) effectively integrating V&V activities into each phase of the M&S development life cycle, such as the articles (Balci and Saadi 2002, Brade 2000, Caughlin 2000, Chew and Sullivan 2000, Wang and Lehmann 2007a), and/or (2) investigating the application of appropriate V&V techniques to support one or more of V&V activities identified in each model development phase, e.g., the related works in (Balci 1998, Love and Back 2000, Carson 2002). The aim of these approaches is to detect different defects as early as possible.

This work however focuses on the identification of the potential interrelationship between defects caused during the M&S development life cycle and presents a generic approach to establishing dependency chains of all associated defects on the basis of structured M&S documents hierarchy, using graph theory. Once a new defect is detected, all of its related defects including the causers and followers, if any, are throughout and in the ordered sequence traceable according to the information indicated in the underlying directed graph. Thus, detection and correction of associated defects can be carried out in a targeted manner, and the required V&V activities are limited to those relevant model documents. This concept can be efficiently combined with other approaches mentioned above, so that various independent model defects are at first detected and then
all of their associated defects are identified and removed consequently.

**GRAPH-BASED DEFECT DETECTION**

Graph theory is normally used as a modelling formalism for representation of a system, such as simulation graphs (Schruben and Yücesan 1993a,b) and event graphs (Schruben 1983, Healy 1993). In addition, there have been many graph theoretic approaches to supporting M&S development, coupled with the capability for graphical interaction, the so-called Graph-Based Modelling Systems (GBMS) (Jones 1990, Geoffrion 1987, Chari and Sen 1998, Chen and Ma 1995). Nance (Nance 1994) summarizes such applications of graph theory. This section describes how graph theory can be used to structure the documents created in the M&S life cycle and to facilitate V&V activities.

**Hierarchical Structure of M&S Work Products**

From the development’s point of view, a simulation study is always initiated by presenting sponsor needs, which can be considered as the first and normally imprecise description of the model to be built. On the basis of the sponsor needs the model should be refined and extended in an iterative manner. During the M&S development life cycle, further work products (also referred to as intermediate products) such as M&S requirements specification, conceptual model, formal model specification, and executable model are successively created and closely associated concerning the contents.

Each model element is a result of some certain work products, and can be also used to develop other M&S products. Consequently, when integrating V&V into an M&S development process, the credibility assessment of a simulation model can be hierarchically structured, and achieved by conducting V&V of each work product, which in turn consists of V&V efforts for every subject contained in the work product.

**Algorithm**

The hierarchical nature of M&S work products can be expressed in terms of graph theory (Bondy and Murty 2008). As Figure 2 illustrates, in a certain M&S development phase all available model documents used to describe different subjects of the work products so far can be considered as vertices and the content dependencies between them as edges of a graph $G(V,E)$, where $V(G)$ is the set of vertices, and $E(G)$ is the set of edges. In addition, if the dependency of each model element, namely the information about all of its direct predecessors and successors is indicated explicitly, a directed graph (or digraph) $G$ is constructed such that for any $(u,v) \in E(G)$, the vertex $u$ (the tail of an ordered edge) represents a precondition for the vertex $v$ (the head of the edge). Since this kind of preconditions is always directed towards successive documents, no directed cycles can be formed. Hence $G$ is also a directed acyclic graph (DAG). Throughout the connected paths of a DAG, it is easy to identify, which model documents in which order influence the contents of another one and vice versa.

![Figure 1: A General Description of the M&S Development Life Cycle](image1)

![Figure 2: Dependencies of M&S Work Products (WP) as a Directed Acyclic Graph (DAG)](image2)

As shown in Figure 1, a work product can be subdivided further into several subjects, each of which represents a specific issue or a model element such as a system of equations, a value, an argument, or an assumption, and can be achieved by application of one or more already available development results. Thus, all model elements are connected with their content dependencies.

To facilitate the dependency tracing of model defects, defects in the context of this paper are classified into two categories: (1) inherited defects and (2) initial defects. An inherited defect is caused by application of erroneous model documents, and is the consequence of other existing defects. Additionally, further defects can also be introduced because of an inherited defect. An
initial defect, however, occurs as a result of human error, and is introduced independently of other defects. It is the origin of all its related inherited defects.

**Algorithm 1: Defect Detection Using DAG**

**Input:** a DAG $\mathcal{G}$, a model defect $d$, the vertex $v$ containing $d$

**Output:** the set of all associated defects of $d$

begin
$v$.checked ← TRUE, queue setOfDefects ← ∅
if $d$.type = INITIAL then
   TOP-DOWN($v$, $d$, setOfDefects)
else
   BOTTOM-UP($v$, $d$, setOfDefects)
end

TOP-DOWN(vertex $x$, defect $d$, queue Defects) begin
foreach $(x, y) \in E(\mathcal{G})$ do
   if $y$.checked = FALSE then
      identify the inherited defect $i$ of $d$ in $y$
      ENQUEUE(Defects, $i$)
   $y$.checked ← TRUE
   TOP-DOWN($y$, $i$, Defects)
end

BOTTOM-UP(vertex $y$, defect $d$, queue Defects) begin
foreach $(x, y) \in E(\mathcal{G})$ do
   if $x$.checked = FALSE then
      identify the causer $c$ of $d$ in $x$
      ENQUEUE(Defects, $c$)
   $x$.checked ← TRUE
   if $c$.type $\neq$ INITIAL then
      BOTTOM-UP($x$, $c$, Defects)
   else
      TOP-DOWN($x$, $c$, Defects)
end

Algorithm 1 shows how the entire associated defects of a newly detected defect can be traced and identified by means of a DAG. For a defect in a certain model document, this detection strategy includes identifying all related defects hidden in the documents which are directly or indirectly connected to the given erroneous model document illustrated as a vertex of a DAG. The defect detection is started with determination of defect type. If the given defect is initial, this means that it has no forerunners but successors, and all of its associated defects are located on the vertices under the current standpoint, then a top-down search is performed, which checks all the connected successive model documents recursively, in order to identify its following inherited defects totally; otherwise (in case of an inherited defect) a bottom-up search is organized, which examines however firstly each model documents upstairs, in order to locate the origin(s) of the given defect, then a top-down search is started individually from each identified origin.

**APPLICATION**

This section presents the application of this graph-based approach within the scope of a V&V framework, the so-called V&V Triangle, which was initially introduced by (Brade 2000), and then consistently extended by the works (Wang and Lehmann 2007b, Rabe et al. 2008).

**Overview of the V&V Triangle**

This framework describes a generalized concept for introducing V&V as part of the model development and life cycle process along which each modelling phase creates a specific and well-structured intermediate product. As Figure 3 illustrates, the V&V Triangle integrates two closely associated parts of V&V activities (model V&V and data V&V) into the underlying exemplary model development process, which defines five modelling phases (depicted as black boxes) and their related intermediate products of Structured Problem Description, Conceptual Model, Formal Model, Executable Model, and Simulation Results. The development phases symbolize sets of activities to transform one intermediate product into its successor.

![Figure 3: The V&V Triangle](image)

The V&V process is organized as a triangle-like matrix. The columns of the matrix represent the V&V main phases, which are associated with the intermediate products; while intersections between the columns and rows split the V&V main phases into V&V sub-phases. During V&V, each intermediate product is examined for internal consistency and completeness with respect to the intended purpose of the model. Subsequently the transformation consistency is checked by pair-wise comparison of all intermediate products. Regarding model V&V, one intermediate product is input to a V&V phase, numbered 1 through 5. Each V&V phase is again split into sub-phases, each with a defined sub-aim to detect the internal defects or transformation...
defects. In each sub-phase numbered as x.1, the absence of internal defects in each particular intermediate product should be demonstrated. For example, in sub-phase 1.1 it should be ensured that the problem description is free of misunderstandings and inconsistencies, and in sub-phase 3.1, a syntax check can be applied to the formal model for comparison of the chosen formalism. In any other sub-phase, the pairwise comparison between the current intermediate product and each previous intermediate product can be performed to confirm the absence of transformation defects. Such as in sub-phases 3.2, 3.3, and 3.4, the formal model could be compared with the conceptual model, the structured problem description, and also the sponsor needs.

With respect to data V&V, two types of data should be distinguished: raw data and processed data. Raw data are obtained directly from different sources, which are in general unstructured and unformatted data. Processed data are, however, created by editing, transforming, or adapting raw data during the modelling process. Thus, data V&V involves credibility assessment of raw data and processed data essential for creating an intermediate product. It should be noted that raw data are usually only relevant for specifying structured problem description and a conceptual model, and are not directly applicable for creating formal models and other succeeding intermediate products. Therefore the associated V&V of raw data for some intermediate products are undefined in this V&V framework (Wang and Lehmann 2007b).

**Combined with the V&V Framework**

The application of the proposed graph-based approach is independent of any model development process, but requires well-structured description of each work products. By clearly defining required contents of the intermediate products, the V&V Triangle provides a guideline for documentation of each step of model development from the V&V perspective. Therefore, the documents available in each development phase are already hierarchically structured as required, and the content dependencies between them can be easily identified as well. Thus, the dependency chains in form of a DAG for tracing the associated defects among the connected model documents are established.

In the V&V Triangle, a different defect classification is applied. Instead of initial and inherited defects, two types are classified: internal defects and transformation defects. Internal defects are defined as inconsistencies, inadequacies, incompleteness, or incorrectness of an intermediate product with respect to rules, facts, assumptions about the real world, while transformation defects occur, when relevant contents of the preceding intermediate products are not completely and consistently reflected by the succeeding products. This defect classification focus on intermediate products specified in the V&V framework and is well-suited for selecting and applying appropriate V&V techniques. The other classification, however, is based on defect propagation and is suitable for tracing the dependencies between related defects. An internal defect can be an initial and also an inherited defect, a transformation defect as well. Because the two defect classifications are used for different purposes, the absence of transformation defects, having the same or similar or even totally modified form. Without identification of the interrelationship between them, all of these associated defects remain isolated and independent of each other. Defects detection and correction can therefore only be carried out in individual cases. As even though the origin is detected, the detection of all its inherited defects can not be followed consequently. As long as some of these defects remain undetected in the model documents, they propagate further.

To resolve this dilemma, the graph-based V&V approach can be used in combination with the V&V Triangle to follow and identify all suchlike related defects. For each defect detected independently within the V&V Triangle, all of the model documents, which are suspect to contain its related defects, are indicated by application of a DAG. As illustrated above, the tracing strategy is based on the determination of defect type (initial or inherited), which requires particular defect analysis activities and cognitions already achieved by using the V&V Triangle. Additionally, the V&V activities specified in the framework are also needed to examine each suspect model document for defect identification. In order to detect associated defects completely, a top-down and a bottom-up search (where applicable) are defined. The two search algorithms work in the style of the common depth-first search (DFS) for digraphs (Manber 1989). Alternatively, breath-first search (BFS) algorithms can also be applied.

**CONCLUSION AND FUTURE RESEARCH**

By means of a directed acyclic graph, this work focuses on investigating the dependencies of associated model defects, and proposes a generic approach to tracing and detecting defects of that kind. To facilitate the dependency tracing of model defects, the classification of initial and inherited defects is introduced. Based on the information indicated in the underlying digraph, detection and correction of associated defects are carried out in a targeted manner, and the V&V activities need to be applied only to those relevant model documents, the unnecessary V&V efforts are therefore avoided. This
approach can be applied to any M&S development process, as long as the work (intermediate) products are well-defined and structured. Due to the hierarchical nature of certain M&S work products, this approach is particularly suitable for tracing and detecting defects in model requirements specification, conceptual model, formal model, and the design specification of executable model.

Our ongoing and future work concentrates on (1) analyzing the factors that influence defect introduction in the M&S development life cycle; (2) developing applicable defect classification for M&S; and (3) proposing general guidelines for selection of well-suited V&V techniques to detect the defined defect types.

Acknowledgment. The authors would like to thank the Federal Office of Defense Technology and Procurement (BWB) for financial support, and Thomas Krieger, Martin Riedl and Johann Schuster for useful comments and suggestions.

REFERENCES


Rabe M.; Spieckermann S.; and Wenzel S., 2008. Verifika-


USING INTEGER LINEAR PROGRAMMING FOR DISCRETE PROBLEM OPTIMIZATION

Jaroslav Sklenar, Valerie Cutajar
Department of Statistics and Operations Research, University of Malta, Msida MSD 2080, Malta
E-mail: jaroslav.sklenar@um.edu.mt
Milan Ceska
Brno University of Technology, Faculty of Information Technology, Department of Intelligent Systems.
Bozetechova 2, 60200 Brno, Czech Republic, E-mail: ceska@fit.vutbr.cz

KEYWORDS
Discrete optimization, Integer programming, Petri nets.

ABSTRACT
We present an application of integer programming that generates optimal examination timetables. The model is based on common assignment problem constraints together with more advanced logical constraints and penalty-based relaxations. In order to minimize the number of constraints we identify dense timetables where the main objective is elimination of clashes and sparse timetables where we spread the exams as evenly as possible. We also mention the possibility of using Petri and related nets to solve this problem.

INTRODUCTION
The main aim of the application is to generate an examination timetable satisfying a number of requirements. The problem is based on real life data and is solved for B.Sc. (Hons.) students at the Faculty of Science of the University of Malta. Students at the Faculty of Science choose two subjects as their main area of studies. Therefore students at the Faculty of Science have different groups and combinations of courses.

DENSE TIMETABLES MODEL
In this chapter we consider a group of students taking some courses during a given time period in a given number of rooms. The input data have the following form:

$S =$ set of students.
$C =$ set of courses.
$D =$ set of examination days. We consider even Sundays and public holidays because they are used as rest days between exams. Exams are usually held on Saturdays; by simple constraints we eliminate exams on Sundays and public holidays.
$H =$ set of Sundays and public holidays.
$R =$ maximum total number of examination sessions that can be held each day. It is not necessarily the number of rooms, because exams can be held both in the morning and in the afternoon. As we try to limit the number of exams per day for each student to 1, the actual time of the exam is ignored in the model.
$A =$ $|C| \times |S|$ matrix with binary values that defines the allocation of students to courses, where $a_{ij} = 1$ represents that the student $j$ is taking a course $i$, 0 represents the opposite.

Model activities are binary variables allocating exams to days:
$E = |D| \times |C|$ matrix with binary values that contains the solution to the exam timetable problem. In particular $e_{ij}$ is 1 if course $j$ is scheduled on day $i$.

We also assume non-negativity of all variables involved in the model. To simplify the notation we assume simple numerical indexes in all sets, so for example the students in the set $S$ are numbered from 1 to $|S|$, similarly for other sets. In the model the actual index sets can be given any values according to the modeling environment used.

Model constraints can be divided into two categories. First there are compulsory constraints that must be satisfied for all acceptable timetables. Other constraints are added to produce a timetable that is attractive for the students. We start by discussing the compulsory constraints. Each course is to be scheduled exactly once. Therefore, the sum of the allocations of a particular course across the days has to be equal to 1:

$$\sum_{j=1}^{|C|} e_{ij} = 1 \quad \forall j \in C$$  \hspace{1cm} (1)

Another compulsory constraint is the maximum number of exams to be scheduled each day:

$$\sum_{j=1}^{|C|} e_{ij} \leq R \quad \forall i \in D$$  \hspace{1cm} (2)

To eliminate exams on Sundays and public holidays we simply force the entries of the matrix $E$ in particular rows to zero:

$$e_{ij} = 0 \quad \forall i \in H, \forall j \in C$$  \hspace{1cm} (3)

Now we describe the optional constraints, which make the timetable more attractive for the students. Due to frequent complaints from students having two exams in the same day, we limit the number of exams per day to 1:

$$\sum_{j=1}^{|C|} e_{ij} a_{jk} \leq 1 \quad \forall i \in D, \forall k \in S$$  \hspace{1cm} (4)

It may happen that the constraint (4) cannot be satisfied for all students and all days. Since the result still may be acceptable if this constraint is not satisfied for some of the students, we introduce penalty variables in usual way:

$$\sum_{j=1}^{|C|} e_{ij} a_{jk} - p_{ik} \leq 1 \quad \forall i \in D, \forall k \in S$$  \hspace{1cm} (5)
where \( p_{ik} \) is the penalty which is the number of additional (with respect to the maximum 1) exams that the student \( k \) has in day \( i \). In the objective function \( p_{ik} \) is multiplied by a cost constant \( K_i \) with large prohibitive value because limiting the number of exams taken by each student in a single day to 1 is in fact the primary objective. Definitely mostly two exams per day are allowed:

\[
p_{ik} \leq 1 \quad \forall i \in D_i, k \in S
\]  

(6)

The timetable also has to be fair to students. So far (5) and (6) don’t distinguish between two clashes for one student or one clash for two different students respectively. Because clashes should definitely be rare, we limit the total number of clashes for each student to:

\[
\sum_{i=1}^{D_i} p_{ik} \leq 1 \quad \forall k \in S
\]  

(7)

To improve the timetable even more, another constraint was introduced in order to impose a day off between any two consecutive exams for each student. Sundays and public holidays were included in the list of days so that these will be taken into consideration:

\[
\sum_{j=1}^{D_i} a_{ik}(e_{ij} + e_{i+1,j}) \leq 1 \quad \forall i \in \{1, ..., D_i - 1\}, k \in S
\]  

(8)

The left hand side in (8) is the total number of exams that a student \( k \) has in the days \( i \) and \( i+1 \). Similarly as for the constraints (4) we introduce penalties, because we have found out that (8) can hardly be satisfied for all students and all days during typical standard exam periods:

\[
\sum_{j=1}^{D_i} a_{ik}(e_{ij} + e_{i+1,j}) - q_{ik} \leq 1 \quad \forall i \in \{1, ..., D_i - 1\}, k \in S
\]  

(9)

In the objective function \( q_{ik} \) is multiplied by a constant \( K_{ik} = K_i + K_k \). We don’t accept exams in three consecutive days:

\[
\sum_{j=1}^{D_i} a_{ik}(e_{ij} + e_{i+1,j} + e_{i+2,j}) \leq 2 \quad \forall i \in \{1, ..., D_i - 2\}, k \in S
\]  

(10)

Again we need a fair distribution of “no rest” cases among students. A simple solution would be a fast increasing nonlinear penalization of the sums of no rest days for all students. To keep linearity and assuming that the total number of no rest days for each student is relatively small, we introduce constraints with penalties on the total of no rest days with increasing upper bounds \( b \) where \( B \) is the largest considered upper bound. Penalties \( q_{ik} \) have been introduced in the constraints (9):

\[
\sum_{j=1}^{D_i} q_{ik} - r_{ik} \leq b \quad \forall b \in \{1, ..., B\}, k \in S
\]  

(11)

In the objective function the penalties \( r_{ik} \) are multiplied by constants \( K_{ik} \) that form a fast increasing sequence, all bigger than \( K_i \) but smaller than \( K_k \). The following is the objective function to be minimized that represents total cost of all penalties:

\[
K_1 \sum_{i=1}^{D_i} \sum_{j=1}^{D_j} p_{ik} + K_2 \sum_{i=1}^{D_i} \sum_{j=1}^{D_j} q_{ik} + K_3 \sum_{i=1}^{D_i} \sum_{j=1}^{D_j} r_{ik}
\]  

(12)

Due to limited space the examples will be presented at the conference. Alternatively contact the authors for the full version of the paper.

**SPARSE TIMETABLES MODEL**

The model described in the previous chapter is suitable for situations where the duration of the examination period is relatively short and thus the primary objective is avoidance of various clashes as described above. In this chapter we consider less tough situations where the duration is not a limiting factor and the main objective is an even distribution of exams for all students in a sense that the gaps between any two adjacent exams are as long as possible. We want to maximize \( u \geq |t_i - t_j| \) where \( u \) is the minimum distance between courses and \( t_i \) and \( t_j \) are any two days when two courses are scheduled such that at least one student takes them both. First we need the times \( t_i \) when the courses are scheduled. They can be linked to the matrix \( E \) of exam schedules by the following constraints:

\[
t_i = \sum_{j=1}^{D_i} e_{ij} \quad \forall i \in C
\]  

(13)

To find out which pairs of courses are taken both by at least one student, we define the clash course matrix \( B \) whose entries are defined as follows:

\[
b_{ij} = \sum_{k=1}^{D_k} a_{ik} a_{jk} \quad \forall i, j \in C
\]  

(14)

In (14) the product is equal to one if student \( k \) takes courses \( i \) and \( j \). \( b_{ij} \) is then the number of students taking both courses. The matrix \( B \) is symmetrical; the diagonal entries that are not used contain numbers of students taking the particular course. The matrix \( B \) is computed from the matrix \( A \), so (14) are not model constraints. For any two courses \( i \) and \( j \) taken both by at least one student, the so-called clash courses, we introduce two mutually exclusive indicator variables \( \delta_{ij} \) and \( \delta_{ji} \) that we shall use to distinguish which of the two courses is scheduled first:

\[
\delta_{ij} + \delta_{ji} = 1 \quad \forall i, j \in C, i < j, b_{ij} > 0
\]  

(15)

Then for each pair of clash courses we use two constraints:

\[
u_g - t_i + t_j + M\delta_{ij} = M \quad \text{and} \quad u_{ij} - t_i + t_j + M\delta_{ji} = M
\]

where \( M \geq 2[D] \) is an upper bound on any possible value \( u_{ij} - t_i + t_j \). If \( t_i \geq t_j \), \( \delta_{ij} \) in the first equality is due to non-negativity of all variables forced to 0, so \( \delta_{ij} \) in the second equality must be 1. That makes the two equalities:

\[
u_g = M - t_i + t_j \quad \text{and} \quad u_{ij} = t_i - t_j
\]

respectively. Similarly in the opposite case. All above pairs of equalities can be expressed as the following constraints:

\[
u_g - t_i + t_j + M\delta_{ij} = M \quad \forall i, j \in C, i \neq j, b_{ij} > 0
\]  

(16)

The following constraints select the smallest time difference \( \nu \) for all pairs of clash courses:

\[
u \leq u_{ij} \quad \forall i, j \in C, i \neq j, b_{ij} > 0
\]  

(17)

We believe that the minimum distance \( \nu \) is not sufficient to assess the quality of a timetable. Together with \( \nu \) we shall use the weighted sum of all distances with weighting coefficients being equal to the number of students taking the particular pair of clash courses. The following constraints define variables \( d_{ij} \) such that \( d_{ij} \leq u_{ij} \) if \( \delta_{ij} = 1 \) and \( d_{ij} = 0 \) if \( \delta_{ij} = 0 \):

\[
d_{ij} \leq M\delta_{ij} \quad \forall i, j \in C, i \neq j, b_{ij} > 0
\]  

(18)
\[ d_y \leq u_y + M(1 - \delta_y) \quad \forall i, j \in C, i \neq j, b_y > 0 \]  \hspace{1cm} (19)

There is no need to impose a lower bound on \( d_y \) because the solver is maximizing the value. In the objective function that has to be minimized, we subtract the following terms from the ones listed already in (12).

\[ K_v + K_a \sum_{i=1}^{C} \sum_{j=1}^{C} b_y d_y \]  \hspace{1cm} (20)

where \( K_v \) is the award coefficient of the smallest distance and \( K_a \) is the award coefficient of the weighted sum of distances. We also note that a minimum distance 2 guarantees that all clash constraints of the basic model are satisfied. In this case we can eliminate all optional constraints of the basic model that will considerably increase the size of soluble models for these not tough cases.

**PROBLEM DECOMPOSITION**

In order to increase the size of models soluble in acceptable time, we applied a problem decomposition approach based on linking models by common courses in a sense that some results of one model will be imposed as constraints in another model. The first decomposition idea was to generate one model regarding the common courses only and to fix these courses in their respective days in all related models. In order that no student will have a clash regarding the common courses, it was decided that only one course per day should be scheduled. The problem arose when it was found that such a model was infeasible since there were more common courses than exam days. It was then decided to split the common courses into three sub-groups by adjacent pairs of years. The first model to be solved involves common courses for years 1 and 2. Then the variables are fixed to solve common courses for years 2 and 3 and to solve the 1st year and the 2nd year timetables, etc.

**CONCLUSION AND FURTHER RESEARCH**

Doing the schedule manually as till now has been very time consuming and usually the results are not very satisfactory. After several weeks of hard work, some students may still have several exams in one day, and therefore, the schedule has to be revised. This means that generating the timetable using integer programming is already much better. Moreover, the sub-optimal decomposition based timetable obtained for several hundreds of students and more than one hundred courses was considered as the significant improvement, although it was impossible to completely avoid clashes. It is important to highlight that some factors have been left out till now. Sizes of rooms and the number of credits that a course is worth were not taken into account when building and running the model. These factors were not considered to be as important as other factors such as clashes and number of rooms available. Also a proper measure for comparison regarding the generated schedule and the manual one was not chosen and tested extensively.

The fundamental problem of integer programming approach is time that limits the size of practically soluble problems. This is well known and one way of tackling this problem is application of various heuristic optimization algorithms. Many papers have been published; see for example (Carter et al. 1996; Burke and Petrovic 2002). We are going to investigate an alternative approach based on the application of Petri and related nets. Integer programming has been used to formulate and solve some Petri Nets problems. See for example (Khomenko and Koutny 2007) or (Melzer and Esparza 1999). Our approach will be the opposite: formulation of the timetable problem by using the Petri nets formalism because there are some similarities. Since the beginning, Petri nets have been used to formulate and solve resource allocation and resource sharing problems – see for example the famous problem of dining philosophers. Timetable problem is nothing else. For example we can identify exams as entities requesting resources – some students and rooms. An exam clash would then be a deadlock in the corresponding Petri net.

On the other hand an attempt to find new approaches and heuristics for efficient solving of extensive integer linear programs and the attempt to apply them in such specific application areas like Petri nets analysis or formal verification are also in the center of our further research.

*This work has been supported by the Grant Agency Czech Republic, the grant No. GA 102/07/0322 “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**


MODEL
REFINEMENT
EMERGENCE AS META KNOWLEDGE:
REFINING SIMULATION MODELS THROUGH EMERGENCE REIFICATION

Daniel David
Remy Courdier
University of La Réunion - EA2525 - LIM - IREMIA
97715, Saint-Denis, La Réunion, France
email: {daniel.david, remy.courdier}@univ-reunion.fr

KEYWORDS
Emergence, Multiagent Simulation, Metaknowledge, Modeling Methodology

ABSTRACT

This paper deals with the important concept of emergence in complex systems and in multiagent simulation. Research in this area yield to several definitions and classifications of emergent phenomena, but only a few of them offer a solution for emergence reification.

As we know, this kind of notion does not have yet, formal definition, if any could be expressed, and we need to progress on the conceptual meaning, leading to more global definitions but allowing to give a general conceptual framework for emergence manipulation.

We define emergence as a metaknowledge controlled by emergence laws, and we present such a framework in which emergence is reified through emergent structures. Within this framework, emergent phenomena can be detected and injected in simulation systems to be manipulated like any other entity.

INTRODUCTION: THE EMERGENCE REIFICATION ISSUE

Simulating complex environments such as natural and social systems is a cross-disciplinary activity. Experts (thematicians such as geographers, biologists, economists) define the models that lie at the core of any simulation: they input the properties and functional descriptions of the system entities. The more the models are refined, the greater the accuracy of the simulations will be: phenomena that arise in the real-world are more likely to arise in the simulated systems. Refining the underlying models is particularly challenging when it comes to emergent phenomena.

Emergence has been an extensively studied concept in the field of complex systems. Work has been done in software engineering (Abbott 2007) and many definitions and classifications have been proposed by the multiagent community (Dessalles et al. 2007, Deguet et al. 2006). Emergence find in Multiagent Simulation (MAS) a very good expressing place whenever emergent phenomena are known to be unilateral or bilateral (Castelfranchi 1998), weak or strong (Dessalles et al. 2007), synchronous or diachronic (Stephan 2002), intrinsic or causal (Boscetti and Gray 2007).

But emergence should not be simply considered as an observation result and emergent phenomena should have a real place in simulations. For example, if we want to simulate a lagoon and its fishes, it would be better to detect potential shoals of hundred fishes that can be formed, and then being able to inject them into the system instead of any of the hundred fishes belonging to the shoals. This is what we call reification. There would be there a significant interest for the system comprehension and in terms of complexity drop during the simulation.

To reach this reification of emergence, we propose a methodology of modeling that makes possible the manipulation of emergent phenomena in multiagent simulations.

EMERGENCE AS META KNOWLEDGE

Defined as “knowledge on knowledge”, metaknowledge offers a greater variety of attitudes and a better way to adapt changes occurring in environments (Luzeaux 1997). This concept is very broad (Pitrat 1990, Kalfoglou et al. 2001) and can be refined so we can consider: the metaknowledge describing knowledge, the metaknowledge on the use of knowledge, the metaknowledge to discover knowledge, and the metaknowledge to manipulate knowledge. This is in full fit with the fundamental concepts of emergence, including the ones of radical novelty (knowledge discovery) and of interdependence levels (use and manipulation of knowledge) (Stephan 2002). According to this, we establish the following property.

Property Emergence is a metaknowledge.

We know that MAS gives us knowledge on studied systems. To reify emergent phenomena, we need to built knowledge and metaknowledge on the simulation in order to detect and learn emergent properties of the simu-
lated systems and to inject into the simulation the properties that emerged also.

**CONCEPTUAL FRAMEWORK FOR EMERGENCE REIFICATION**

**Introspection Process** consists in creating and expressing knowledge and metaknowledge on the simulation in order to detect phenomena that do emerge.

To detect the emergent phenomena that appear in a simulation, we need to know the set (or a set) of all the facts that are occurring (or have occurred, in case of post-simulation analysis) in the MAS. These facts are related to agents behaviors and to their environment. They are function of space, time, communication, and, more generally, of the different entities and the different metrics present in the MAS. This set of facts can be built by observing the interactions of the agents and the evolution of the environment(s) in which they evolve. We define this set $\mathcal{K}$ as follows:

$$\mathcal{K} = \{\text{facts}\}$$

This set represents the useful knowledge (for emergent phenomena analysis) that can be studied on the MAS. So, in the MAS context, we can define $\mathcal{K}$ as follows:

$$\mathcal{K} = \mathcal{K}_A \cup \mathcal{K}_E$$

where $\mathcal{K}_A$, which gathers the facts produced by agents, is defined by:

$$\mathcal{K}_A = \{\text{influences, perceptions, interactions}\}$$

and where $\mathcal{K}_E$ is the set of all the facts produced within the environment.

The emergent phenomena are defined through the study of this set of facts $\mathcal{K}$. Emergent phenomena are defined through an analysis of $\mathcal{K}$, that leads to establish relations between facts. We define as follows an extensible set of functions $\mathcal{R}_E$:

$$\mathcal{R}_E = \{f : \mathcal{K}^n \rightarrow \text{boolean}\}, \quad n \in \mathbb{N}$$

Each function of this set detects if combinations of parameter facts define an emergent phenomenon. We call these functions **emergence revelators**. Using them, we can now define the $\mathcal{P}_E$ set of emergent phenomena:

$$\mathcal{P}_E = \{f \in \mathcal{R}_E / f = \text{true}\}$$

Once emergent phenomena have been detected in simulations, we need to give them “life”.

**Intercession Process** consists in the injection into the MAS of the emergent phenomena that have been detected during the introspection process.

The detection of emergent phenomena sometimes yields to the definition of entities that will directly influence agents behaviors in the MAS. These entities manifest themselves through two kinds of emergent elements that we define hereafter: **emergence agents** and **interposition elements**.

An **emergence agent** is an agent that runs on a MAS platform. It thus evolves in the same environment as all other agents of the system and interacts with them through the mechanisms of influence and perception offered by the platform. Several emergence agents can be created to reify the same phenomenon.

An **interposition element** is a modification of one or several environments. It changes (as appropriate by altering, improving, restricting, etc.) the perception or influence mechanisms associated with one or more agents. These two elements are controlled by emergent metastructures that we call $m_{SE}$, which are themselves governed by laws of emergence. Emergence laws are all the elements of the set $\mathcal{L}_E$ defined as follows:

$$\mathcal{L}_E = \{f : \mathcal{P}_E^n \rightarrow \mathcal{S}_E\}, \quad n \in \mathbb{N}$$

where $\mathcal{S}_E$ represents the set of all the emergent structures. Each emerging structure is defined by a tuple $<\text{emergent metastructure, emergence agents, interposition elements}>$.

In the shoal of fish example, the shoal of fish will be represented by an emergence agent. The fishes that constitute the shoal of fishes may continue to evolve in their environment, but will have their influences and perceptions changed by elements of interposition controlled by the $m_{SE}$ corresponding to the shoal of fish. The individual fishes will not have anymore to exchange messages to find collectively the best direction for moving because the shoal will take this decision for all of them: the complexity drop is important.

Our approach supports the dynamism of emergent phenomena, especially their volatility. That is, when an emergent phenomenon is no longer observed, its corresponding emergent structures are deleted from the MAS. Indeed, the functions in $\mathcal{P}_E$ (the ones that activated the corresponding laws in $\mathcal{L}_E$) are no longer defined.

**Emergence Reification** consists in doing the complete cycle of the introspection and intercession processes.

Figure 1 shows a summary of our proposal, with the “inside simulation” aspect on one hand and the “conceptual vision” on the other hand for the complete cycle of emergence reification: introspection and intercession. It is important to notice that because emergent structures can have their own behavior, they create new
knowledge that can be added to the set of facts. And this contributes to refine the knowledge on the MAS. With this conceptual framework, thematicians have a new role to play during this cycle of design: thinking about the \( R_E \) and \( L_E \) sets that appear in the processes of introspection and intercession. This is consistent with what happens in the real world: thematicians do not have any innate knowledge on emergent phenomena; it is only from observation that they have learned to recognize, characterize or naming them. But the emergent phenomena that has been detected are a kind of new knowledge, and it is important to add this new knowledge to the global knowledge of the system. This will yield to a better understanding of the simulation system itself and to the discovery of new emergent phenomena.

CONCLUSION

In this paper, we focused on the emergence issue and on its representation in MAS. We consider that we need to improve the way this concept is taken into account in simulations, and we proposed and described a conceptual framework for the detection and injection of the different kinds of emergent phenomena in simulation. Actually, it is the analysis of the knowledge on the simulation that allows us to reify emergent phenomena. That is why our key proposition is the definition of emergence as a metaknowledge on the MAS: it leads to the definition of introspection and intercession processes that we introduced to reach the emergence reification.

In future works, we will use the conceptual framework we described in this paper to improve a multiagent application of energy simulation under development on our platform GEAMAS-NG. The goal is to take emergence into account as soon as we start to design simulation agents while keeping a clear separation between initial behaviors and emergent ones. This would improve the agents by giving to them the possibility of reasoning on themselves and so on emergent phenomena.

REFERENCES


Kalfoglou Y.; Menzies T.; Althoff K.; and Motta E., 2001. Meta-knowledge in systems design: panacea... or undelivered promise? The Knowledge Engineering Review.


Luzeaux D., 1997. Learning knowledge-based systems and control of complex systems. 15th IMACS World Congress.


ANALYSIS AND SIMULATION OF THE LEG OF AN HEXAPOD ROBOT FOR REMOTE EXPLORATION.

J. Torres 1, G. Romero 2, J. Gomez-Elvira 1, J. Maroto 2

1 Centro de Astrobiología INTA/CSIC, Ctra de Torrelodones a Alcalá de Henares, km 4, 28850 Torrelodones de Ardoz, Madrid Spain  
2 ETS Ingenieros Industriales, Universidad Politécnica de Madrid (UPM), Jose Gutierrez Abascal, 2, 28006 Madrid, Spain  
E-mail addresses: torrestj@inta.es (J. Torres), gregorio.romero@upm.es (G. Romero), gomezj@inta.es (J. Gómez-Elvira), joaquin.maroto@upm.es (J. Maroto).

KEYWORDS
Hexapod robot, walking machines, rover, remote exploration, simulation.

ABSTRACT

The locomotion system is determined by the terrain conditions. The aim of this paper is to introduce the characteristics and simulation of a hexapod legged robot that can be easily used for exploration of abrupt and harsh terrains, like the Rio Tinto environment.

A walking robot seems like the best option for this kind of terrain. Some of the advantages are that they do not need a continuous terrain, they have less problems with sliding and they also have greater capacity to overcome obstacles as they produce less harm to the environment that the scientist wants to explore on the contrary when faced with mechanical design they present a design challenge, also in the static and dynamic analysis problem of a legged robot, there is a high complexity that has to be taken into account.

This paper shows how to easily cope with the analysis of a hexapod robot movement based on a design developed by the Center of Astrobiology INTA-CSIC for operation in Rio Tinto (Huelva - Spain).

WALKING MACHINES

Walking robots (fig. 1) are well suited for unstructured environments and abrupt terrains having some drawbacks like movement coordination, speed and power consumption.

![Walking Machines](image)

Figure 1. a). Katharina (Fraunhoffer Inst.), b). Scout I (MIT).

In order to choose the best configuration the state of the art in walking machines has to be reviewed. Walking machines are usually found classified by their number of legs ranging from eight to one. Walking robots with six legs and four legs are the more frequently built, there exists multiple designs based on the choice of actuators, dimensions and design implementations like articulated bodies giving high numbers of active degrees of freedom (Song and Waldron 1989).

With new technologies advancing an increasing market has begun with microrobots, alone or in communities that can work cooperating between each other are appearing, this kind of robots have the disadvantage that they perform precise tasks that do not deal with high payload capacity. Last but not least bipeds and one legged machines are reaching the markets specially bipeds which simulate and perform human actions as shown in the following figures.

![Walking Machines](image)

Figure 2. a). Honda human robot, b). Ole Vrije Universiteit.

As it has mentioned before walking machines look like a good option when it deals with harsh terrains, they can adapt better to abrupt terrains and be less harmful to the environment they are in, rover wheels usually damage the terrain we want to analyze or explore. Some of the walking machines summarized before are feasible for remote exploration of harsh terrains. But the design had to avoid “mechanical complexity” which means low reliability, high power consumption and large control problems (Song and Waldron 1989).

When dealing with a robot design of this complexity, the design will be based on the design requirements. These will depend on the type of scientific missions goals and these will have to be clearly specified by the scientific group for the design engineers to begin. When a design of this type starts it will vary depending on the scientific payload that needs to be carried, the type of environment that wants to be explored and the range of area that needs to be covered. These will affect the design materials, type of actuators, mechanics and power consumption that needs to be used. These will give an estimate dimensions of our robot that will also help us with the decision of the robot configuration.

The Center of astrobiology having in mind all the design specifications and mission requirements involved and taking
into account the number of different instruments that will like to be placed in the robot has designed a six legged robot having in mind the abrupt terrain that it will work in ‘Rio Tinto’. (Huelva, SouthWest of Spain) (Bruhn 1992). The Tinto River has shown a large scientific interest due to its special characteristics from a biological and geological point of view. The study of such environment is characterized by its hard conditions, beside the need of performing some of the experimentation in situ. This forces some exceptional exploration methods, like the use of robots. The robot will develop its activities in a rocky terrain with dry and wet areas, like those shown in the following figure.

Figure 3. ‘Rio Tinto’ terrain.

This rover must be designed having in mind that will transport a scientific payload and it should be able to achieve as much location as possible in order to collect information of different river zone. One of the elements more sensible to the terrain conditions is the locomotion system because since it should make feasible any displacement along the river. The locomotion system is determined by the terrain conditions. The aim of this article is to introduce a six legged robot with low power consumption and maximum payload capacity. The following paragraphs will summarize the first trade off calculations of the design and its first analysis and simulations.

**HEXAPOD DESIGN**

Based on a design requirement of 40 kilograms of payload, and in order to minimize mechanical complexity the final design chosen after analyzing several configurations, concluded in a hexapod shaped rover powered by linear DC actuators (Torres and Pomares 2002, Barrientos and Peñin 1997). These type of actuators were chosen for its high force outputs as we are dealing with an approximate 80 kilograms robot. The robot has a design configuration like the one shown in figure 3 (Waldron et al. 1984). The main function of the robot hexapod box is to carry the payload. It has to be large enough to fit all of the payload and robust to withstand the forces and moments due to the walking motion and possible collisions with objects. The hexagonal design is to allow more space between the legs when they perform the rowing motion (Bruhn 1992). The number of legs were chosen as a compromise between complexity and stability (Song and Waldron 1989), and six seemed like a good compromise.

With these six leg configuration the walking motion could be performed as follows: the robot can stand in three legs in a triangular configuration meanwhile the other three legs are in the air. The three legs on the floor can perform a rowing motion forward and when they reached the maximum motion, the legs in the air can come down and the same procedure will follow, with these movements we can avoid having the 18 actuators acting at the same time (Torres and Pomares 2002).

![Figure 4. Dimensions of the prototype design.](image)

**LEG DESIGN**

When dealing with any design the desired result from a mechanical point of view is its simplicity in manufacturing, also simplicity during mounting and with repair, this design was thought primary to be highly robust (Gere and Timoshenko 1990). The linear DC motors chosen will allow large movements of the legs but will act as part of structural members of the mechanical design (McGuee and Orin 1976).

Each of the legs has three independent degrees of freedom, the linear actuators controls two degrees of freedom The third degree of freedom is controlled by a DC gear motor. Below is a detailed sketch of the leg and the minimum and maximum positions of all the actuators involved. The maximum angles and positions that it can move on will depend on collisions, large force or moments acting and unstable positions.

![Figure 5. Attachment and function of the actuators.](image)

The materials will be aluminum hollow bars, which are light but have high mechanical strength and to assure smooth movements they will pivot in bearings.

**REACTIONS**

After analyzing the robot mechanical design, the first trade off calculations that will improve and change the design have to be made. The first step to start the calculations is to find
out what are the reactions \( R \) in each of the individual legs taking into consideration all of the weights the structure and actuators are going to withstand at all of the possible leg configurations. The procedure is to find the moments and forces about a point in the hexapod leaving a system of three equations and three unknowns, these equations will be listed here, and solved using Matlab\textsuperscript{®}. The reactions are found in the worst case assuming that the hexapod is standing in three legs, which are performing the rowing motion. The other three legs do not have a ground load reaction but contribute with their weight and displacement configuration. The following figure explains graphically the loads involved. Weight loads are shown only at one leg labeled ‘6’. Each leg is suffering the same represented weights as leg ‘6’, but different \( R_2, R_4 \) and \( R_6 \) will be obtained depending on the configurations.

For the verification of ground reactions a static Nastran\textsuperscript{®} force analysis will be performed to validate the results obtained from Matlab\textsuperscript{®}.

Also from these Nastran\textsuperscript{®} analysis, materials and different design configurations will also be analyzed, but this will not be dealt in this article and from the program the reactions will be taken to compare our theoretical results, these is later shown below in results.

![Figure 6. Ground reactions.](image)

The resulting equations are reduced to three due to the addition of forces and moments at point ‘0’. Due to the direction of the vertical forces acting, ‘z’ distances do not appear in the equations. This three resulting equations are a function of the next shown variables which are calculated under a loop so all possible configurations are taken into consideration. Legs ‘1’, ‘3’ and ‘5’ are kept at fixed positions so the program has enough memory to iterate the different configurations of the three legs which have ground reactions.

\[
\begin{align*}
\alpha_{2,4,6} & = -5^\circ: 30^\circ \quad (1) \\
\beta_{2,4,6} & = 85^\circ: 120^\circ \quad (2)
\end{align*}
\]

As the three legs are rowing at the same time it can be assumed that:

\[
Y_2 = Y_4 = Y_6 = -20^\circ: 20^\circ \quad (3)
\]

The resulting equations can be summarized to the following system of equations:

\[
\begin{bmatrix}
1 & 1 & 1 \\
Y_2 & Y_4 & Y_6 \\
X_2 & X_4 & X_6
\end{bmatrix} \begin{bmatrix}
R_2 \\
R_4 \\
R_6
\end{bmatrix} = \begin{bmatrix}
-W_T \\
M_x \\
M_y
\end{bmatrix}
\]

Equations I. Reaction equations.

where

\[
W_T = \text{Addition of all weights, box + legs + actuators (‘F’ equation)}
\]

\[
M_x = \text{All the independent terms in the moment ‘1’ equation for all of the legs acting} \quad (’Y’ \text{ distance to point ‘0’})
\]

\[
M_y = \text{All the independent term in the moment ‘2’ equation for all of the legs acting} \quad (’X’ \text{ distance to point ‘0’})
\]

\[
X_2 = ‘X’ \text{ distance from the reaction force ‘R2’ to point ‘0’}
\]

\[
Y_2 = ‘Y’ \text{ distance from the load reaction ‘R2’ to point ‘0’}
\]

**LEG FORCE ANALYSIS**

Due to the number of unknowns the analysis of each of the robot legs has to be broken into separate sections in order to be able to calculate these unknowns.

The calculations will be performed for leg ‘2’ which as its illustrated in figure 6 withstands the greatest ground reactions, varying ‘\( \alpha_2 \), ‘\( \beta_2 \)’ and ‘\( \gamma_2 \)’.

The following figure will illustrate the first cut section that will be calculated.

![Figure 7. First section analysis.](image)

In figure 6, we have reduced the number of equations and unknowns to three unknowns and three equations, with these we can begin to calculate the first reaction forces.

The three equations involved are the summation of forces in ‘x’ and ‘z’ directions and taking moments about point ‘0’ which is located where ‘\( R_2 \)’ and ‘\( R_6 \)’ are acting.

\[
\sum F_x = R_z + R_2 - m_z \cdot g + F_{\text{act}} \cdot \cos(\phi_{\text{act}}) - m_{\text{act}} \cdot g \quad (5)
\]

\[
\sum F_z = R_x + F_{\text{act}} \cdot \sin(\phi_{\text{act}}) \quad (6)
\]
\[
\sum M = R_z \cdot X_{d_{0z}} - m_{l_z} \cdot g \cdot X_{d_{0z}} + F_{axz} \cdot \cos(\phi_{axz}) \cdot X_{d_{ax}} + F_{axz} \cdot \sin(\phi_{axz}) \cdot Z_{d_{axz}} + m_{axz} \cdot g \cdot X_{d_{axz}}
\]  
\[
\text{Equation 7}
\]

\[
\begin{bmatrix}
1 & 0 & \sin(\phi_{axz}) \\
0 & 1 & \cos(\phi_{axz}) \\
0 & 0 & \sin(\phi_{axz}) - \cos(\phi_{axz})
\end{bmatrix}
\begin{bmatrix}
R_z \\
R_x \\
F_{axz}
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
m_{l_z} \cdot g - R_2 + m_{axz} \cdot g
\end{bmatrix}
\]  
\[
\text{Equation 8}
\]

Equations II. Force equations in first section analysis.

We can continue with the second section cut, with the value of the reactions in the hinge we start solving for the next unknowns shown which can also be reduced to three equations and three unknowns that can easily be solved.

The equations involved are the summation of forces in ‘x’ and ‘z’ directions and taking moments about point ‘0’ which is located where ‘R_{xz}’ and ‘R_{xz}’ are acting (eqs. III).

\[
\sum F_x = R_{xz} + R_{x} - m_{l_z} \cdot g + F_{axz} \cdot \sin(\phi_{axz}) - F_{axz} \cdot \cos(\phi_{axz}) - m_{axz} \cdot g
\]  
\[
\text{Equation 9}
\]

\[
\sum F_z + R_{x} + F_{axz} \cdot \cos(\phi_{axz}) - F_{axz} \cdot \sin(\phi_{axz})
\]  
\[
\text{Equation 10}
\]

\[
\sum M = -m_{l_z} \cdot g \cdot L_z + F_{axz} \cdot \sin(\phi_{axz}) + F_{axz} \cdot \cos(\phi_{axz}) \cdot L_z \cdot \sin(\alpha) - F_{axz} \cdot \sin(\phi_{axz}) \cdot L_z \cdot \cos(\alpha) + R_2 \cdot L_z \cdot \cos(\alpha) + R_x \cdot L_z \cdot \sin(\alpha) + F_{axz} \cdot \sin(\phi_{axz}) \cdot L_{ax} - m_{axz} \cdot g \cdot X_{d_{axz}}
\]  
\[
\text{Equation 11}
\]


\[
\begin{bmatrix}
1 & \cos(\phi_{axz}) \\
0 & \sin(\phi_{axz}) \\
0 & \cos(\phi_{axz}) \cdot L_z \cdot \sin(\alpha) - F_{axz} \cdot \sin(\phi_{axz}) \cdot L_z \cdot \cos(\alpha)
\end{bmatrix}
\begin{bmatrix}
R_z \\
R_x \\
F_{axz}
\end{bmatrix}
= 
\begin{bmatrix}
-R_x + F_{axz} \cdot \sin(\phi_{axz}) \\
m_{l_z} \cdot g - R_x + F_{axz} \cdot \cos(\phi_{axz})
\end{bmatrix}
\]  
\[
\text{Equation 12}
\]

Equations III.b. Matrix force’s equations in second section analysis.

Figure 8. Second section analysis.

In the last section cut, shown in figure 9, the vertical bar is attached with bearings so it can be assumed that is simply supported withstanding the forces shown, due to the resultant number of unknowns and equations it has to be reduced to an easily solved case. Each of the acting forces can be reduced to a force and a moment acting at the ends of the vertical bar. The moment force appears because the attachments have an offset in the ‘x’ and ‘z’ direction.
In this section, we have more unknowns than equations so we have to apply a simple method in which we can reduce it to three unknowns, for these we can assume that every force that acts in the bar is equal to the sum of a force plus a moment, from the Roark’s book of formulas for stress and strain (Young and Budynas 2001) we find the following:

In figure 10 we have the four unknowns ($R_{z1}$, $R_{z2}$, $R_{z3}$, and $R_{z4}$) and four known forces acting ($F_{act1}$, $F_{act2}$, $R_{z1}$, and $R_{z4}$), that we can apply to them the above reduction case. The distance at which the force is acting is important. We have forces acting in two points at which a from figure 10, will be equal to ‘0’ and ‘$L_3$’.

For ‘$a=0$’ , the force acting is from actuator ‘1’, the force $F_{act1}$ in the ‘x’ and ‘y’ direction is equal to:

$$F_{act1_x} = F_{act1} \cdot \sin(\phi_{act1})$$  \hspace{1cm} (13)

$$F_{act1_y} = F_{act1} \cdot \cos(\phi_{act1})$$  \hspace{1cm} (14)

Applying the equations of figure 10 at ‘a=0’ and ‘$L=L_3$’, we obtain the following values for point ‘1’.

$$R_{z1} = \frac{P_2}{L} \cdot (L-0) = P_2 = F_{act1}$$  \hspace{1cm} (15)

$$R_{x0} = P_2 \cdot \frac{0}{L} = 0$$  \hspace{1cm} (16)

$$R_{y0} = \frac{F_{act1_x} \cdot x_{dist}}{L_3}$$  \hspace{1cm} (18)

$$R_{z0} = \frac{F_{act1_y} \cdot z_{dist}}{L_3}$$  \hspace{1cm} (19)

$$R_{z0} = -R_{z1}$$  \hspace{1cm} (20)

Applying the equations for the reaction forces ‘$R_{zj}$’ and ‘$R_{z1}$’, obtained in earlier calculations we obtain:

$$R_{y0} = \frac{P_2 \cdot L - (L_2 - L_1)}{L_2} = 0$$  \hspace{1cm} (21)

$$R_{z2} = P_2 \cdot \frac{L_3}{L_2} = P_2 = R_{z1}$$  \hspace{1cm} (22)

$$R_{x0} = \frac{R_{z1} \cdot x_{dist}}{L_3}$$  \hspace{1cm} (23)

$$R_{z0} = \frac{R_{z1} \cdot z_{dist}}{L_3}$$  \hspace{1cm} (24)

$$R_{y0} = -R_{z1}$$  \hspace{1cm} (25)

$$R_{z2} = -R_{z1}$$  \hspace{1cm} (26)

By adding the following forces we can find the reaction forces in the vertical bar:

$$R_{y0} = R_{y0} + R_{z1}$$  \hspace{1cm} (27)

$$R_{y0} = R_{y0} + R_{z1}$$  \hspace{1cm} (28)

$$R_{y0} = R_{y0} + R_{z1} + R_{z1} + R_{z1} + R_{z1} + R_{z1}$$  \hspace{1cm} (29)

$$R_{y0} = R_{y0} + R_{z1} + R_{z1}$$  \hspace{1cm} (30)

Equations IV. Vertical reaction forces.

RESULTS

The equations here presented were programmed in Matlab and validated with Nastran, a finite element program. Nastran took the design a step further by taking into account real dimensions and materials that were going to be used (Song et al. 1985).

In this way it can be verified that the previous calculations were the reaction force in the legs from the weight of the payload and the different geometrical configurations that the robot will have to encounter matched our calculations.

Also it is a way to check that the materials and components selected are able to resist these reaction loads that will appear during the walking motion and irregular configurations.

The steps followed are clear through the article.

1.- The first step was to analyze the ground reactions; this was a first approach in order to know the magnitude of the load that each leg will be carrying. By breaking in sections we can easily find the reactions and forces acting in all legs. Knowing the forces that are going to act in our robot legs, we can now choose the right actuator that suits our application and the materials that will resist these loads.
A closer look at the forces and moment of each leg for a value in each section, the equations are programmed in Matlab© for the different configurations that the robot will encounter and a second review of the actuators and material has to be performed.

![Figure 11. Reactions Matlab© results.](image1)

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Maximum Force (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{F1Y}$</td>
<td>1200</td>
</tr>
<tr>
<td>$F_{F2Y}$</td>
<td>1200</td>
</tr>
<tr>
<td>$R_{x}$</td>
<td>1300</td>
</tr>
<tr>
<td>$R_{y}$</td>
<td>600</td>
</tr>
<tr>
<td>$R_{z1}$</td>
<td>1300</td>
</tr>
<tr>
<td>$R_{x1}$</td>
<td>1400</td>
</tr>
<tr>
<td>$R_{z2}$</td>
<td>1400</td>
</tr>
<tr>
<td>$R_{x2}$</td>
<td>2000</td>
</tr>
<tr>
<td>$R_{z3}$</td>
<td>600</td>
</tr>
<tr>
<td>$R_{x3}$</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 1. Maximum force values for each reaction.

Knowing the ground reactions we can proceed to the section cuts, were we obtain the rest of the desired forces, they are shown in the above plots and are summarized in the table. The forces will depend on the elongation of the actuators, with the desired elongation required in the design and the maximum limit force that we want to have we can have a trade off of the actuators as the force will constraint the size of the actuators. Very heavy actuators are not desired as they are part of the structure and will have to be redesigned according to the new weight and dimensions.

2.- Validation of the results a structural program like Nastran© and conclusion of the design and components involved.

![Figure 12. Reaction Nastran© analysis for one configuration.](image2)

Here a static subcase is shown of our robot it shows a configuration that we had found previously as of high acting forces, we can see that the point were the leg has the weight on its own shows the maximum value of 418 N. If we take a look at figure 11, ground reactions versus ‘X’ distance, we find a maximum value for the leg standing in its own with a maximum value of 500 N, this value is when the leg is in a position nearest to the center body.

This is consistent with the results found theoretically also this involves materials and real geometry which gives a more closer value to the one we are going to encounter in reality. This model was also simulated for all of the theoretical cases analysed, here only one static case is shown.
CONCLUSIONS

The analysis of a complex design such as this, has to be taken with a lot of care and having very clear in the design what your goals are. Start off information as what your design is meant for, what load do you want to carry and what environment has the design to face will limit the development and analysis of the robot.

This paper was meant to explain the equations around a hexapod robot configuration but also to clarify and teach a person facing with a similar design problem how to cope with the analysis by approaching the problem by section and steps. These calculations will probably have to be done cyclical every time new actuators or new payload is selected in order to achieve a trade off between weight, characteristics and budget.

Figure 13. Hexapod robot design developed by the Center of Astrobiology INTA-CSIC.

An important issue in this design will be validation with the real model, right now the first prototype is in the Robotics lab of the Center of Astrobiología (fig. 13) and before starting any field work the fist tests of mobility based in this paper have to be performed. Also dynamic movement and pre programmed trajectories will be a task to be performed in the following months and will be dealt in future articles.

REFERENCES


BIOGRAPHY

JOSEFINA TORRES received her BS engineering degree in 1998 and her master degree in Aerospace engineering in 2000 from Parks College (USA). She has worked as an assistant professor at Parks College from 1998 to 2000 in the Mechanical engineering department. She is currently working at the Robotics and Planetary Exploration group of the Center of Astrobiology in Madrid were she works in structural analysis and dynamic modelling in research projects together with ESA and NASA. Currently she is also working in her Mechanical engineering PhD at the Technical University of Madrid in Spain (UPM). She has published more than 10 technical papers and has been actively involved in over 10 research projects.

GREGORIO ROMERO received his Mechanical Engineering and Doctoral degrees from the UNED (Spain) in 2000. He got his PhD Degree from the Technical University of Madrid in Spain in 2005 working on simulation and virtual reality, optimizing equations systems. He has worked as Assistant Professor at the Technical University of Madrid in Spain (UPM) since 2001. He is developing his research in the field of simulation and virtual reality including simulation techniques based on bond graph methodology integrating computer graphics and virtual reality techniques to simulation in real time. He has published more than 35 technical papers and has been actively involved in over 20 research and development projects and different educational projects.

JAVIER GOMEZ-ELVIRA received his PhD degree in Aeronautical Engineering in 1981, were he began to work at the National Spanish Institute of Aerospace Technology (INTA), he has participated in the calibration of different airships and in an international certification group inside the Joint Aviation Authorities. Since he started at INTA he has been involved in the development of different instrumentation for different satellites and the technical supervision of the first generation of HISPASAT satellites. Actually he is responsible for the Laboratory of Robotics and Planetary Exploration of the Center of Astrobiology where different projects on space instrumentation are been developed in collaboration with ESA and NASA. He has published over 30 technical papers and has been actively involved in over 20 research and development projects.

JOAQUÍN MAROTO received his Control Engineering and Doctoral degrees from the Madrid Politehnic University in 2000 and 2005. He has been Assistant Professor at the Technical University of Madrid in Spain (UPM) since year 2003. His main activities and research interests are mainly focused on the field of simulation, computer graphics, virtual reality and machine vision. His main contribution is in the field of distributed virtual environment generation and the generation of immersive systems. He has published over 30 technical papers and has been actively involved in over 25 research and development projects.
LOCALISATION OF JOINT ROTATION CENTRES FOR 3D HUMAN MOTION SIMULATION

Martin Pruzinca and Hildegard Köhler and Annika Wörmer
Institute for Algorithms and Cognitive Systems
Universität Karlsruhe
Kaiserstr. 12
76131 Karlsruhe,
Germany
pruzinca@ira.uka.de, kohler@ira.uka.de, woerner@ira.uka.de

KEYWORDS
Model analysis, Model transfer, Intelligent simulation environments, Approximation techniques, Signal processing.

ABSTRACT
Procedures dealing with the simulation and the analysis of human motions are based on a kinematic model of the human anatomy. In case the analysis of individual attributes is required, precise modelling of the human, realised by the adaption of a predefined model to the one of a subject, is essential. We propose a motion based method for estimating the location of joint rotation centres based on a reduced set of 3D markers attached to the body. The presented method assumes that markers attached on a segment of the body will not change their distance to related joints of the segment and all trajectories will be located on spherical paths around the related joint. By the evaluation of several marker data, the joint rotation centre can be localized with an accuracy of few millimetres.

INTRODUCTION
Application fields for human models range from medical rehabilitation and sports to entertainment and product design like in the automotive industry. Most models in use need to be manually adapted to certain anatomic features of a test person. Hence, an automation of the process of individualization of an adaptive model by providing a swift opportunity for automatic capturing of anatomic features is preferable.

This adaption can be done top down by regarding the whole body structure and adapting the model to it e.g. by a global fitting or by a bottom up approach adapting the position of local joint coordinates to the kinematic of the test person. A global fitting method has already been presented by the authors in (Köhler et al. 2008). To complement this top-down approach, the presented method deals with the second scenario, trying to estimate the kinematic joint centres of body segments, which is equal to the centre of rotation of the joint. The basic assumption is that markers attached to a segment will not change their distance to the joint. To estimate the joint centres, a local coordinate system is determined for each body-segment. The calculation of the joint centres has to be done in a 2D or 3D coordinate system, depending on which type of joint is examined.

ALGORITHMIC APPROACH
Estimation of Coordinate System
To analyse the marker trajectories related to a predefined joint centre, the independence from other motions than those originated from the wanted joint is essential. This results in a transformation of marker points on the connected segments from the global coordinate system into a local system based around that joint. At first an appropriate set of at least three markers on the connected body segments is chosen. As the markers are located on top of the skin, they are influenced by soft tissue movements. To reduce this movement, the determination of appropriate markers is done by examining their change of distance over the time between markers on the same segment and by choosing those with minimal variance. With this set of markers the local coordinate system is created. The origin can be defined by a random marker of the set. The x-axis is specified by the vector from the origin to one marker point on the same body segment. The y-axis is defined as parallel vector to a perpendicular on the x-axis crossing through a marker point on the other body segment.

Estimation of Joint Centre in 2D
As there are different joint types in the human anatomy, we have to take different motion profiles into account. To differentiate between joint types, the approach is presented for saddle joints, with a 2D motion profile and for ball joints with a 3D motion profile. The trajectory of markers around joints like the saddle joint only varies in two dimensions as the joint is assumed to have one degree of freedom. The
trajectories can be approximated through a circle, as shown in figure 2. This assumption is feasible since the human body does not have any translative joints and hence the distance between a point on a connected segment and the joint do not vary over the time of the motion. We translate the global 3D marker coordinates into local 2D coordinates, by projecting the z-coordinates to the plane. This is possible as in case of a joint with one degree of freedom the deviation of 3D points on the z-axis will only be caused by the volume of the underlying body segment. But even in case of a combination of saddle and pivot joint, e.g. at the elbow, this assumption holds.

![Figure 1: Transformed trajectories of two wrist markers and estimated joint centre.](image)

The task is to estimate the rotation point G as the wanted joint rotation centre which has the minimal variation in its distance to all markers. In dependency of time t and the marker trajectory m, we have to minimize the distance formula with respect to G as follows:

\[
h(m_i(t)) = (x_i(t) - G_x)^2 + (y_i(t) - G_y)^2 - r_i^2
\]

where \( r \) is the radius and \( G \) the centre of the cycle. \( r \) is the distance between the marker \( i \) and \( G \). Defined over the whole motion the term is:

\[
H(m_i) = \begin{bmatrix} h(m_i(t_0)) \\
h(m_i(t_k)) \end{bmatrix}
\]

Because all markers are moving around the same centre point we can formulate the concentric term for all markers over the whole motion:

\[
\min \sum_{i} H(m_i) = \min \sum_{i} \sum_{t} \sqrt{(x_i(t) - G_x)^2 + (y_i(t) - G_y)^2 - r_i^2}
\]

For solution of this optimisation problem a non-linear implementation of the least square algorithm by Coleman (Coleman and Li 1996) is used.

**Estimation of Joint Centre in 3D**

The trajectories of segment points around joints with more than one degree of freedom, e.g. the ball joint, show 3D changes. To locate those joints the before shown formulas has to be adapted to 3D. At first we translate the global coordinates into a 3D local coordinate system. The approximation of the trajectory of each marker is done with a sphere as can be seen in figure 2. Analogue to the 2D case we build a concentric optimization term depending on all trajectories of the markers and the time of examination:

\[
\min \sum_{i} H(m_i) = \min \sum_{i} \sum_{t} \sqrt{(x_i(t) - G_x)^2 + (y_i(t) - G_y)^2 + (z_i(t) - G_z)^2 - r_i^2}
\]

![Figure 2: Transformed trajectories of elbow marker and estimated elbow joint.](image)

**RESULTS**

For this study marker based 3D motion capture data with different people as described by (Boesnach et al. 2006) are analysed. The placing of the markers varies with each person and recording session as the markers are manually placed on the body according to a rough guideline. The used 3D marker data are recorded with an optical recording system. In the data-set the recorded marker-points are completely labelled so an exact mapping of recorded coordinates to markers is given. The motions sequences examined are executed by 7 different people accomplishing standard kitchen work. This includes a walking sequence of 4 steps followed by putting dish into dishwasher with both hands (figure 4). The subjects performed around 15 to 20 iterations. The method has been applied on the two elbows, shoulders and knee joints. Related markers are listed in table 1.

<table>
<thead>
<tr>
<th>Estimated joint</th>
<th>Markers for local coordinate system</th>
<th>Markers used for estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elbow left</td>
<td>LSHO, LAEL, LWTS</td>
<td>LWTS, LWPS, LFRS</td>
</tr>
<tr>
<td>Elbow right</td>
<td>RSHO, RAEL, RWTS</td>
<td>RWTS, RWPS, RFRS</td>
</tr>
<tr>
<td>Shoulder left</td>
<td>LAEL, LSHO, C7</td>
<td>LAEL, Elbow joint left</td>
</tr>
<tr>
<td>Shoulder right</td>
<td>RAEL, RSHO, C7</td>
<td>RAEL, Elbow joint right</td>
</tr>
<tr>
<td>Knee left</td>
<td>LHIP, LKNE, LTIB</td>
<td>LTIB, LANK</td>
</tr>
<tr>
<td>Knee right</td>
<td>RHIP, RKNE, RTIB</td>
<td>RTIB, RANK</td>
</tr>
</tbody>
</table>

**Table 1: Marker set for the estimation of joint rotation centres described by Boesnach (Boesnach et al. 2006).**
Hence no ground truth exists the results have been evaluated by calculating the average distance between the marker trajectories and the calculated spheres. For each trajectory $m(t)$, we used the term:

$$Q = \frac{1}{N} \sum_{i=1}^{N} |m(t) - \hat{m} - r|$$

We achieved an average deviation over all subjects and iterations between trajectories and calculated spheres between 0.5 – 1.5mm as shown in figure 5.

In figure 3 the maximal and minimal deviation between the estimated spheres and markers trajectories for each joint and subject over all iterations is shown. Larger deviations occurred especially at the shoulder joints of test person one and four. These outliers were mainly caused by strong motions of the muscles and skin next to the used markers. We analysed what minimal amount of motion data is required to get accurate results.

Hence we implemented a threshold value as a minimal distance between the used trajectory points and their predecessor in time. By maxi

**CONCLUSION**

The presented method achieved precise results with a deviation in the lower millimetre domain. For now it is sensitive to errors due to strong soft-tissue and muscle movements near the markers. In the future we will add a weighting function to filter out those outliers by penalizing large distances between marker positions to their predecessor in time.

**ACKNOWLEDGEMENT**

This work was supported by the grant from the Ministry of Science, Research and the Arts of Baden-Württemberg.

**REFERENCES**


MANAGING THE RESOLUTION OF SIMULATION MODELS

Gábor Lenese  
Department of Telecommunications  
Széchenyi István University  
Egyetem tér 1.  
H-9026 Győr, Hungary  
e-mail: lenese@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, model resolution management,  
information and communication technology systems,  
business process systems

ABSTRACT

A novel approach based on inflation and deflation is proposed for managing the resolution of simulation models. Different methods are proposed for manual or automatic deflation. An example is given how a topology description language can be extended to support the inflation/deflation concept. Dynamic management of the model resolution is introduced using the method called inflate-the-next and also two of its possible improvements.

INTRODUCTION

Resolution of the Models

Discrete-Event Simulation (DES) is a widely used method for the performance analysis (Jain 1991; Pidd 1991) of Information and Communication Technology (ICT) systems and Business Process (BP) systems. There is 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 or on computer grid. The 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 both modelling work and computing time. It seems to be logical that the most efficient resolution of a model is a compromise between these two requirements. Time decomposition method described in (Muka and Lenese 2007) may help the modeller to choose the relevant systems to be modelled, to manage the appropriate model resolution and to support the decision about parallel and sequential simulation. The appropriate level of resolution may be found by applying a set of transformations to the different parts of the model. However, it is many times desirable that the resolution of the model can be adjusted dynamically. Why? On the one hand, the modeller may not have enough information for determining the optimal resolution in the model building stage – the situation may improve later during experimenting with the model. On the other hand, the different experiments performed on the model may require different resolution of the different parts of the model.

In this paper, it is shown how the resolution of the simulation models can be made volatile giving freedom to the modeller to set the appropriate resolution for each part of the model for each experiment individually.

The remainder of this paper is organised as follows: first, the idea of inflating and deflating is presented, second, some methods for deflation are presented, third, an example is given how a topology description language can be extended to support the inflation/deflation concept, fourth, the dynamic management of the resolution is introduced.

THE IDEA OF INFLATION AND DEFLATION

Before presenting the inflation/deflation concept, we briefly consider the possible elements of the models we deal with.

Modelling Concept

In order that our results can be widely used in the world of modelling and simulation of information and communication technology (ICT) and business process (BP) systems, we do not intend to make any unnecessary restrictions on the models. We believe that it is a rational expectation that a contemporary modelling and simulation environment should give the possibility of hierarchical system description. We call the key element of the hierarchical description Compound Module. A compound module may contain further compound or simple (that is: not compound) modules and other elements depending on the modelling and simulation environment.

Changing the resolution by inflation/deflation

To support the flexibility of the resolution of our models, we simply add a new attribute (some modelling systems call it parameter) named Resolution to all the compound modules. This attribute is of an enumerated type and may take the values either Deflated or Inflated. These are the states of the compound module. How do we interpret these states? If a compound module is inflated, its internal structure as well as the operation of its parts are modelled in detail. (In the same way as it is done without the resolution attribute.) If it is deflated, we ignore its internal structure and operation and imitate its behaviour for the outside world by a simpler algorithm that acts similarly but of course not completely the same as the original compound
module. As a compound module may contain several elements – even compound modules of arbitrary hierarchical levels –, we expect that the simulation of a compound module in the deflated state requires much less computing power and models the real world system less precisely than in its inflated state. The advantage of the flexibility is that the modeller may decide experiment by experiment how much precision is necessary in the modelling of the given compound module. In this way we must pay the price (in computing power) of the necessary precision only.

The concept above means that at a given hierarchical level the modeller may decide about all the compound modules to be inflated/deflated as he/she wishes, but if a compound module is deflated, no more decision can be made about the contained compound modules. However, if a compound module is inflated, some of its contained compound modules may still be deflated, as due to encapsulation the compound module “knows” nothing about them. Thus, it is meaningful to define a new expression: a compound module is fully inflated if it is inflated and all the contained compound modules are fully inflated. (Illustrated in Fig. 1.)

**Figure 1.** CM1 is: a) Fully Inflated, b) Inflated, c) Deflated

**Manual or automatic deflation**

In the simplest case, the modeller has to write two descriptions (codes) for each compound module: one for its inflated state and one for its deflated state. We call this approach *manual deflation* and it has the following advantages: the modeller has full control over the model in both states and no extra features are required from the modelling and simulation system. The price is that the modeller has to make two models for each compound module.

It is possible that the modeller prepares the model of the compound module for the inflated case only. We will show different methods that may be suitable for *automatic deflation*, where the model of the compound module in the deflated state is generated automatically. The advantage of this approach is that the modeller does not have to work on the model of the compound module in the deflated state. This approach requires either support from the modelling and simulation system or some extra work from the modeller, however this extra work probably may be automated quite well. The details depend on the methods used for automatic deflation.

**METHODS FOR DEFlation**

The methods presented here are based on our previous research results in modelling and simulation. Even though the essence of these results will be briefly summarized (as space permits) the reader is encouraged to read the original papers to get deeper understanding of the methods used here.

**Substitution by Statistical Interfaces**

The basic idea of this solution has its roots in the Statistical Synchronisation Method (SSM) invented by György Pongor (Pongor 1992) and further developed under the name SSM-T by Gábor Lencse (Lencse 1998a).

![Figure 2. An OIF - IIF Pair](image)

*Statistical Synchronisation Method*

SSM is a Synchronisation Method for Parallel Discrete Event Simulation (PDES). A synchronisation method is responsible for keeping causality between the virtual times of the segments. More well-known methods for this task are the conservative and the optimistic methods – more information on them can be found in (Fujimoto 1990). Now, we summarize the essentials of SSM in a nutshell. Similarly to other parallel discrete-event simulation methods, the model to be simulated — which is more or less a precise representation of a real system — is divided into segments, where the segments usually describe the behaviour of functional units of the real system. The communication of the segments can be represented by sending and receiving various messages. For SSM, each segment is equipped with one or more input and output interfaces. The messages generated in a given segment and to be processed in a different segment are not transmitted there but the output interfaces (OIF) collect statistical data of them. The input interfaces (IIF) generate messages for the segments according to the statistical characteristics of the messages collected by the proper output interfaces. (See Figure 2.) The segments with their input and output interfaces can be simulated separately on separate processors, giving statistically correct results. The events in one segment do not have the same effect in other segments.
as in the original model, so the results collected during SSM are not exact. The precision depends on the partitioning of the model, on the accuracy of statistics collection and regeneration, and on the frequency of the statistics exchange among the processors.

Deflation with Statistical Interfaces
It is natural that SSM can be applied in the following way: the contents of the compound module will create one segment and all the other parts of the model will create the other segment. In this way SSM is applied at the boundaries of the compound module. Let us consider the message routes that lead from inside the compound module to the outside world. Let us wait until all the OIFs collect enough data for their first statistics. They send the statistics to the proper IIF that will regenerate the message flow for the outside world. Now the simulation of the inside part of the compound module may be stopped and the IIFs will still sustain the message flow. Of course, this approach works only in the steady state of the modelled system. This is what the original SSM can be used for.

Non-steady State Behaviour
Here, SSM should be replaced by SSM-T. Besides the one that was mentioned before, we have published a number of papers on the different issues of this method, such as dealing with its statistics collection methods (1998b) applicability criteria (Lencse 1999a) and statistics exchange control algorithm (Lencse 1999b). Using these results, it seems to be possible both to automatically stop the detailed simulation of the inside parts of the compound module and to sustain the message flow, and also automatically restart it when necessary. This topic can be a subject of further research.

Substitution by Flow-based Methods
By flow-based methods we mean the Traffic-Flow Analysis (Lencse 2001) for ICT systems and the Entity Flow-Phase Analysis (Lencse and Muka 2006) for BP systems.

Traffic-Flow Analysis
The Traffic-Flow Analysis (TFA) is a simulation-like method for fast performance analysis of communication systems. TFA uses statistics to model the networking load of applications.

In the first part, the method distributes the traffic (the statistics) in the network, using routing rules and routing units.

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.

As TFA is a less well-known method, it has only one partial implementation, which is a part of the ImiNet network expert system (Elassys 2008).

Entity Flow-Phase Analysis
The Entity Flow-phase Analysis has been derived from TFA. This derivation is based on the formal similarity of the ICT and BP models. EFA uses the same two phase method 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 the aforementioned paper.

An implementation of EFA is planned as an extension for the ImiFlow system (Elassys 2008).

Deflation with TFA or EFA
On the basis of our results presented in (Lencse and Muka 2007) it is trivial that the detailed simulation of the internal parts of a compound module can be replaced by TFA for ICT systems and by EFA for BP systems. The different modelling methods still should work together. Thus the manual deflation seems to be quite simple. For the automatic deflation, we need a method to automate the transformation of the detailed ICT or BP models to TFA or EFA models, respectively. This topic seems to be a very interesting and promising research area.

TOPOLOGY DESCRIPTION LANGUAGE SUPPORT
We show an example how a network topology description language can be extended to support the inflation/deflation concept. We selected the NeD language\(^1\) (Varga and Pongor 1997) for this purpose. The EBNF description of the compound module is the following:

```
moduledefinition ::= 
  module compoundmoduletype
  [ paramblock ]
  [ gateblock ]
  [ submodblock ]
  [ connblock ]
endmodule [ compoundmoduletype ]
```

Now we extend it in the following way:

```
moduledefinition ::= 
  module compoundmoduletype
  [ paramblock ]
  [ gateblock ]
  [ submodblock ]
  [ connblock ]
  [ deflatedblock ]
endmodule [ compoundmoduletype ]
```

\(^1\) The EBNF grammar of the NeD language can be found in the User Manual of OMNeT++ (http://www.omnetpp.org).
deflatedblock ::= 
deflated: 
[ paramblock ]
[ gateblock ]
[ submodblock ]
[ connblock ]
enddeflated

In simple words, the EBNF description above means that after the optional keyword “deflated:” the modeller may give the description of the compound module for the deflated case by the same way (using the same types of blocks) as he/she could describe the contents of the compound module for the inflated case. (And the description for the inflated case remained the same as it was originally.) As for the Resolution attribute, the NeD grammar requires no modification, the modeller can provide a parameter with this name. Let us see the EBNF of the parameter description:

paramblock ::= 
parameters: { parameter , , , } ;

parameter ::= 
  parametername
  | parametername : const [ numeric ]
  | parametername : string
  | parametername : bool
  | parametername : char
  | parametername : anytype

If we do not insist on the enumerated type with the values Deflated, Inflated, it seems to be a very simple solution to implement these values by logical values of the Boolean type: false and true respectively.

RESOLUTION MANAGEMENT

Now, we introduce a highly cost effective solution for simulation of large and complex systems. Let us suppose that we have the detailed model of the system containing many elements that would require a huge cluster of processors for simulation experiments, but we have only limited computing capacity.

Inflate-the-next Method

At the highest hierarchy level of the model, we divide the set of modules into two parts: the set of inflated modules and the set of deflated modules. The set of inflated modules should contain all the modules that we focus on at a given step of the given experiment. And all the rest of the modules are deflated and put in the deflated set of modules. In a simple case, we assign the two sets of modules to two separate processors for execution. (Of course, an arbitrary number of processors may be used.) Between the processors we may use any synchronisation methods for PDES mentioned before. For the next step of the experiment, we deflate the contents of the inflated set of modules and move them into the deflated set of modules and we select to inflate the next set of modules from the deflated set of modules and inflate them and put them into the inflated set of modules. This is why we call this algorithm inflate-the-next. The experiment ends when all the relevant sets of modules have been inflated and executed. See Figure 3.

![Figure 3. Inflate-the-next Method](image)

The Order of Selection

We may have two approaches:

- If we choose a set of compound modules to be inflated with a minimum number of connections to the rest of the modules then hopefully we will get relatively precise results for the inflated set.
- If we choose a set of compound modules to be inflated with the maximum number of connections to the rest of the modules then the results of execution can be better used to model them when they are deflated.

Using these considerations, the inflate-the-next algorithm can be improved in the following way. All the experiments should contain two phases: in the first phase we use the approach with maximum number of connections to improve the precision of the model, in the second phase we use the approach with the minimum number of connections to increase the precision of the simulation results.

The Size of the Inflated Set

On the one hand, the size of the inflated set is limited by memory and computing power; on the other hand, possibly all the modules with intensive communication with each other should be put into a single inflated set of modules. This idea has at least two advantages:

- It eliminates the problem of decreasing precision that would be caused by putting modules with intense communication with each other into separate inflated set of modules.
• It speeds up the parallel simulation of the two sets by decreasing the number of messages between them.

This consideration should be used together with the previous one.

Directions of Future Research

The ideas presented in this paper seem to be promising. Both the research of further algorithms for deflation and the investigation of the presented ones can be a hot research topic. The resolution management algorithm and its improvements should be tested and refined.

CONCLUSIONS

We have introduced a concept based on inflation and deflation for maintaining the optimal resolution of simulation models.

We have presented different methods for manual and automatic deflation.

We have shown a way how the grammar of a typical topology description language can be modified to support the inflation/deflation concept.

We have given a resolution management method called inflate-the-next as well as two possible improvements for this method.

We conclude that both the topic of resolution management and the methods we presented deserve further research.

REFERENCES


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 has been working 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 Infrastructure 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 has been working 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 Infrastructure Systems at the Széchenyi István University of Győr.
MODELLING METHODS IN INDUSTRY
MODELLING OF COMPACTION AND RESIDENCE TIMES IN CONTINUOUS COOKING APPLICATION

Miika Uusitalo and Timo Ahvenlampi
Systems Engineering Lab., University of Oulu, Finland
e-mail: timo.ahvenlampi@oulu.fi

KEYWORDS
Continuous cooking, compaction, modelling

ABSTRACT

In this study, the compaction of the chip column and residence times in the continuous Downflow Lo-Solids\textsuperscript{TM} cooking application are investigated using modelling. The chip pressure, Kappa number, chip velocities and chip volume fractions in the digester are modelled and used in the residence time calculations in order to enhance controllability of the digester. The proposed models are compared with the models found in the literature. The models are tested also in the grade changes and several production rates. Industrial data is used in the evaluation.

1 INTRODUCTION

The compaction of the chips can be a big problem for the control of the digester. The uniform packing degree is an essential factor affecting on the movement of the chip column. Chip column’s fluent movement enables the better handling of the residence times, the temperature and alkali treatment and further the quality of the blow line pulp. By calculating the wood and liquor balances, the changes in the packing degree can be indicated. Flow rates and velocities have been also calculated for the chips and liquors in the digester. The detection and elimination of any changes or disturbances in the balances, velocities and other process variables enable a more even control of the whole process.

The quality of the pulp is normally defined using online Kappa number measurement, which is located in the blow line. The kappa number indicates the residual lignin content of the pulp and the main variables affecting it are the temperature and alkali profiles in the digester and residence time in cooking process and the wood species. The control of the kappa number is very important part of the continuous digester process. The quality of the pulp has a major effect on the quality of the final paper. Proper pulp quality control is very important to pulp production in the entire fibre line (Leiviskä 2000).

In this study, the compaction of the chip column in the industrial continuous cooking application is researched. The chip velocity in the digester determines the residence time in the system. The residence time has a major effect to the Kappa number. Due to this, the compaction is an important factor to the control of the process and quality of the end product. If the chip column compaction is known in the digester, the process disturbances are easier to identify and rectify.

The compaction phenomena has been modelled in several papers, see e.g. (Härkönen 1987), (Lammi 1996) and (Lee and Bennington 2004). In these papers, laboratory data has been mainly applied. There are not many papers dealing with industrial data. Rantanen & Kortela (Rantanen and Kortela 2006) modelled compaction and residence times in the conventional continuous cooking application. In this paper, compaction is studied in industrial Downflow Lo-Solids\textsuperscript{TM} continuous cooking application. The aim in this study has been to enhance Scandinavian birch and pine models to be suitable for the Downflow Lo-Solids\textsuperscript{TM} cooking application.

In the following section, the studied process is presented. In next section, the models found in the literature and proposed models are shown. Next results are presented and finally discussion and conclusions are done.

2 PROCESS

Studied application is a Downflow Lo-Solids\textsuperscript{TM} (Marcoccia 1996) cooking process (Fig. 1). The chips are impregnated in the impregnation vessel (I-12) and in the first zone (D1) of the digester. Between upper extraction and cooking circulation there is a counter-current zone (D2). In this zone, black liquor is displaced with cooking circulation liquor which temperature and alkali concentration are high. The lignin is mainly removed in the comparatively long co-current cooking zone (D3). At the bottom of the digester there is a short washing zone. Softwood chips mainly consist of pine chips with a small amount of spruce chips. Hardwood chips consist mainly of birch chips.

The effective alkali concentrations of the white liquor (A1 in Fig. 1), of digester feed circulation liquor (A2), of two black liquor (A3, A5) and of cooking circulation (A4) are measured. The white liquor is added to the im-
pregnination vessel’s feed circulation, to the digester’s feed circulation and to the cooking circulation. The sulphide concentration of the white liquor is measured. Temperatures are measured from the impregnation vessel (T1 and T2), the liquor circulations (T3 and T6), the extractions (T5 and T7) and from the heating steam (T4) at the top of the digester. A temperature profile from the top of the digester to the cooking circulation is constructed emphasizing the measured temperatures suitably. The temperature profile from the cooking circulation to the blow line is based on the temperature of cooking circulation.

3 MODELLING OF THE COMPACTION

The physical phenomena inside the industrial continuous cooking digester are a complex mixture of chemical reactions and transport phenomena. The missing measurement data leads to a large number of uncertain parameters. Due to this, the modelling and identification of the cooking process is a challenging task.

The compaction and the velocity of the chips in the digester changes during the transport from the top of the digester to the blow line. The main effects are caused by the density differences between the penetrated chips and free liquor and delignification. The delignification softens the chips and the chip column compaction increases. These changes are modelled in this study and new model parameters are proposed.

Harkonen built a chip pressure model in his study (Härkönen 1987). In the equation 1, is shown the modified chip pressure model by (Saltin 1992)

$$\frac{\partial p_c}{\partial h} = (p_c - p_l) \epsilon_c \frac{\rho_c - \rho_l}{D_{\text{digester}}} + \frac{p_c}{D_{\text{digester}}} \frac{\partial p_l}{\partial h}, \quad (1)$$

where $p_c$ is a chip pressure, $\epsilon_c$ is volume fraction of the chips, $\rho_c$ is density of the penetrated chips, $p_l$ is density of the free liquor, $\mu$ is friction coefficient, $D$ is the digester diameter, $h$ is the height of the area $g$ is the gravity coefficient and $p_l$ is a liquor pressure.

The sign $\pm$ depends on the directions and relative value of the flows of the chips and liquor in the digester.

Harkonen (Härkönen 1987) used the Ergun equation based liquid flow resistance model in the compressed chip column.

$$\frac{\partial p_l}{\partial h} = R_1 \left[ \frac{\epsilon_c^2}{1 - \epsilon_c} \right] v + R_2 \left[ \frac{\epsilon_c}{1 - \epsilon_c} \right] v^2, \quad (2)$$

where $R_1, R_2$ are species related coefficients and $v$ is relative velocity between chips and liquor.

Table 1: Species related coefficients in equation (2) presented in the literature (*, (Härkönen 1987)), (**, (Lammi 1996))

<table>
<thead>
<tr>
<th>Species</th>
<th>$R_1$</th>
<th>$R_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scandinavian pine, *</td>
<td>4600</td>
<td>$3.9 \cdot 10^6$</td>
</tr>
<tr>
<td>Scandinavian birch, **</td>
<td>-17</td>
<td>$1.7 \cdot 10^6$</td>
</tr>
<tr>
<td>Eucalyptus camaldulensis, **</td>
<td>3.2</td>
<td>$3.7 \cdot 10^6$</td>
</tr>
</tbody>
</table>

Harkonen (Härkönen 1987) built empirical equation also for the volume fraction of the chips. The volume fraction depends on the chip pressure and delignification rate i.e. Kappa number.

$$\epsilon_c = k_0 + \left( \frac{p_c}{10^4} \right)^{k_1} \left( k_2 \ln(K) - k_3 \right) \quad (3)$$

where $k_0, k_1, k_2$ and $k_3$ are species related coefficients and $K$ is Kappa number.

Table 2: Species related coefficients in equation (3) presented in the literature (*, (Härkönen 1987)), (**, (Lammi 1996))

<table>
<thead>
<tr>
<th>Species</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scandinavian pine, *</td>
<td>0.356</td>
<td>0.59</td>
<td>0.139</td>
<td>0.831</td>
</tr>
<tr>
<td>Scandinavian birch, **</td>
<td>0.37</td>
<td>0.64</td>
<td>0.151</td>
<td>0.697</td>
</tr>
<tr>
<td>Eucalyptus camaldulensis, **</td>
<td>0.409</td>
<td>0.56</td>
<td>0.148</td>
<td>0.675</td>
</tr>
</tbody>
</table>

The volume fraction of the free liquor, $\epsilon_l$, can be calculated using the following equation

$$\epsilon_l = 1 - \epsilon_c \quad (4)$$
Proposed model parameters

In the construction of new model parameters, the above-mentioned models are utilised. The original parameters were not suitable for the studied continuous cooking application. Although, the same models structures were applied in the study. The proposed parameters for chip pressure model are presented also in the work by Uusitalo (Uusitalo 2008).

The new model parameters for the chip pressure model are shown in Table 3.

Table 3: Proposed parameters for the chip pressure model

<table>
<thead>
<tr>
<th>Species</th>
<th>$R1$</th>
<th>$R2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scandinavian pine</td>
<td>4600</td>
<td>$3.9 \cdot 10^5$</td>
</tr>
<tr>
<td>Scandinavian birch</td>
<td>1500</td>
<td>$2.9 \cdot 10^6$</td>
</tr>
</tbody>
</table>

As can be seen, the Scandinavian pine parameters are not changed. Although, the Scandinavian birch parameters are modified for the studied process.

The proposed parameters for the volume fraction model are presented in Table 4. The Scandinavian pine parameters are modified only slightly. The Scandinavian birch parameters are changed evidently, because the original parameters were not suitable for the studied continuous cooking application. The modification was based on mill experiments and identification using measurement data from the plant’s automation system and modelling results. The test chips were collected from the chip bin conveyor. The volume fractions were calculated for the both species (hardwood and softwood) using volume and mass of the test chips. The Scandinavian birch model was optimised using the mill tests and the residence times in the digester. The Kappa numbers used in the calculations are modelled using modified Gustafson’s (Gustafson et al. 1983) Kappa number model. Regression method was used in the identification.

Table 4: Proposed parameters for the volume fraction model

<table>
<thead>
<tr>
<th>Species</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scandinavian pine</td>
<td>0.356</td>
<td>0.59</td>
<td>0.135</td>
<td>0.78</td>
</tr>
<tr>
<td>Scandinavian birch</td>
<td>0.393</td>
<td>0.6</td>
<td>0.085</td>
<td>0.4603</td>
</tr>
</tbody>
</table>

4 RESULTS

In this section, the modelling results and comparisons are presented. The results are shown for the hardwood (HW) and softwood (SW) species in changing production rates. The modelling and simulation is done in

Matlab.

Densities

In Figs. 2-3, penetrated chip and free liquor densities are presented. The free liquor densities of the different flows are based on the laboratory analysis in the studied plant. The free liquor densities of the digester zones are calculated using mass balances. The penetrated chip densities are based on the chip densities, free liquor densities and yield in the zones (the dissolved wood material by delignification and carbohydrate degradation in the studied zone).

Chip pressures

The chip pressures are calculated using the equations 1-3. Also the calculated penetrated chip and free liquor densities are applied in the chip pressure models. In
Figs. 4-5, the chip pressure simulations are presented. In Fig. 4, the Scandinavian birch parameters from Lammi (Lammi 1996) and in the case of Scandinavian pine parameters from (Härkönen 1987) are applied, see Table 1.

![Chip bed pressures](image)

Figure 4: Chip pressures in the digester.

The modelling results using proposed parameters (see Table 3) are presented in Fig. 5.

![Chip bed pressures](image)

Figure 5: Chip pressures in the digester using proposed parameters.

**Chip and free liquor velocities**

The chip and free liquor velocities are calculated using the volume fraction models, volume of the chips or free liquor and area of the digester in the current zone. The chip and free liquor velocities in the digester zones using proposed models are presented in Figs. 8-9. The velocities are applied in the calculation of the residence times.

**Residence times in the digester**

In Fig. 10, the production based and calculated residence times are compared with Harkonen’s (Härkönen 1987) Scandinavian pine and Lammi’s (Lammi 1996) Scandinavian birch parameters. The production based and calculated residence times are compared with the proposed parameters in Fig. 11.

**5 DISCUSSION**

In this study, the compaction and residence times in the continuous Downflow Lo-Solids™ cooking application is researched. The compaction cannot be measured in the digester. Therefore models are needed to get the compaction information. The compaction is modelled using chip pressure. The chip pressure is not the same as liquid pressure. Chip pressure can deviate greatly from the liquid pressure. The contact force between chips is transferred via chips to the next contact point. This force cannot be measured and therefore chip pressure is defined. The chip pressure is average of the contact forces over the studied surface. (Härkönen 1987) The chip pressure in the top of the digester is modelled using the difference between the chip and liquor levels, chip density and chip volume fraction. The chip pressures in the other parts of the digester are achieved using the models for the densities, the chip volume fractions, chip and free liquor velocities and liquor flow resistance.

Because the original volume fraction and chip pressure models were not suitable for the studied process new parameters were modified for the models. The softwood models were close to the optimal and only slight modification was applied to the softwood volume fraction model. The hardwood models needed more modification. The new parameters were calculated using laboratory tests and empirical modelling using measurements from the plant’s automation system. The laboratory tests were done for the liquor densities in the digester’s input and output flows and for the chip volume fractions before the cooking process.

Based on the laboratory tests the volume fraction in the top of the digester was found to be too low in the hardwood case when using Lammi’s (Lammi 1996) parameters. Because of that the volume fraction model...
parameters for the hardwood case were modified for the studied application. The volume fraction modelling results are shown in Figs. 6 and 7. As can be seen from Fig. 6, when using Lammi’s parameters the volume fraction of the hardwood in the upper part of the digester is very low (about 0.38 in zones D1 and D2). The mill experiments indicated bigger fractions and the volume fraction models were modified using these results. Also the chip density (481 kg/m$^3$) of the hardwood is bigger than with softwood (400 kg/m$^3$), which indicates bigger volume fractions for the hardwood at the top of the digester. In Fig. 7, is shown the chip volume fractions with proposed models. Also the softwood model was changed little. With these parameters the volume fraction of the hardwood is bigger than with softwood in the upper part of the digester (after zone D1). In the counter current zone (D2), the volume fractions increase only slightly, due to the counter current liquor flow.

The chip pressures using parameters shown in the literature and proposed parameters are presented in Figs. 4 and 5. Although, the results seem to be nearly the same, the inner modelling results (density difference and flow resistance parts in equation (1)) are in very different levels.

The modelled residence times are shown in Figs. 10 and 11. In Fig. 10, the calculated and production based residence times are presented using model parameters by Lammi and Harkonen. The residence times by proposed parameters are shown in Fig. 11. As can seen, the modelling results are closer to the production based ones.
The validated models are accurate although the modelling of the compaction and residence times is very challenging. Some of the models will be implemented into the plant’s automation system.

6 CONCLUSIONS

The chip pressures, chip volume fractions, chip velocities and Kappa number are modelled using data from the continuous cooking application. New parameters are proposed to the models found in the literature. Especially in the case of hardwood pulping the models had to be changed for the studied Downflow Lo-Solids\textsuperscript{TM} type digester. Using these models it is possible to calculate the residence times in the digester zones. The calculated residence times are compared with the production based residence times. Industrial data is used and good modelling results are achieved.

![Image of residence times in the digester using models with Härkönen’s (Härkönen 1987) Scandinavian pine and Lammi’s (Lammi 1996) Scandinavian birch parameters.](image)

Figure 10: Residence times in the digester using the models with Härkönen’s (Härkönen 1987) Scandinavian pine and Lammi’s (Lammi 1996) Scandinavian birch parameters.

![Image of calculated chips residence times vs. production based residence times in the digester.](image)

Figure 11: Residence times in the digester using proposed model parameters.


ACKNOWLEDGEMENTS

This research study is funded by The Academy of Finland. The data is from Oy Metsä-Botnia Ab. The authors would like to thank the partners for the special knowledge and process data provided.

REFERENCES


INFLUENCE OF THE GEOMETRY OF THE WORKPIECE ON CONTACT PRESSURES IN FRICTIONLESS INDENTATION PROCESSES

Marta M. Marin
Ana M. Camacho
Eva Rubio
Miguel A. Sebastián
Department of Manufacturing Engineering,
National Distance University of Spain (UNED)
E-28040 Madrid,
Spain
E-mail: mmarin@bec.uned.es

KEYWORDS
Manufacturing and Engineering, Industrial processes, Numerical methods, Nonlinear

ABSTRACT

Indentation processes change the geometry and properties of the workpiece as a result of compression forces. Although usually indentation has been used as a method for determining the material hardness, in this work indentation will be considered as a manufacturing process because of the great advantages that can be obtained.

In this study the contact pressures have been analyzed under the influence of the geometry of the billet. The dimension of the punch will remain constant while the dimensions of the billet are changed. The indentation process has been studied assuming ideal conditions consisting of plane strain ones. Friction between the workpiece and the punch has been neglected assuming that a good lubrication has been realized. Different cases have been analyzed by the Finite Element Method, obtaining the profiles of contact pressures. This numerical technique allows defining complex geometries and boundary conditions and also a more realistic material response. All the cases have been solved for a strain hardened material. As a result it has been observed the contact pressures with regard to the height and width of the workpiece. Thus, the higher the height and the width, the higher the contact pressures. Additionally it has been observed where the maximum values take place, and from what width the response become equal for the rest of heights.

In this paper indentation processes have been studied as a manufacturing process. Traditionally, indentation has been associated with obtaining hardness, one of the main mechanical properties of materials. Often studies on indentation are basically tests in which the hardness of a material is obtained and results depend on the shape of the punch and the used scale (Chen et al. 2006; Kucharski and Mróz, 2007; Minh-Quy, 2008).

Hardly ever indentation processes have been studied as compression processes using their advantages to obtain a useful product. However, indentation processes can be used for producing parts of excellent mechanical properties with minimum waste of material (Sebastián and Camacho, 2007).

If the indentation process is compared with other compression processes it is observed that the result of the forces to carry out each operation is lower. In this sense, it is an interesting advantage because it allows us to choose powerless equipments. This is a new approach of forming process that presents high flexibility because it is possible to obtain different parts simply developing different CNC programs and the use of general purpose machines reduces costs.

Manufacturing a product essentially requires energy and information. The energy allows shaping materials through the information described (Camacho et al. 2007) so the optimization of the process will require as much information as possible.

Considering all the information necessary to carry out indentation processes is very complex. For this reason, this study has been focused on analyzing the influence of the geometry of the workpiece on the contact pressures. The dimension of the punch will remain constant throughout the paper, while the dimensions of the billet are changed. As a starting point of this study a previous work has been used with the aim of developing a more complete analysis (Marín et al. 2008b).
In this paper the contact pressures have been chosen as main results. (Pereira et al. 2008) The workpiece has been considered a rectangular billet. The geometry of the workpiece has been changed both in height and width. Friction between the punch and the workpiece has been considered null.

METHODOLOGY

Parameters

At first, it can be considered a plain strain problem. A rectangular geometry of the punch is considered, where $B$ is the width. This dimension remains constant throughout the whole study.

And the workpiece is considered a rectangular billet as well, but its dimensions ($w$, $h$, width and height respectively) change throughout the analysis (see Figure 1).

![Figure 1: Geometry of punch and billet](image)

These changes have been carried out by geometrical relations. The geometrical relations have been obtained from other studies (Marín et al. 2008a). In these studies a shape factor has been defined and the dimensions of the billet are changed according to this shape factor. It has been defined by a geometric relation with the dimensions of the punch.

As it can be observed in the Figure 1, the problem presents symmetry, and this allows considering a half of the original workpiece and punch in the model in order to simplify the calculations. In all the cases, the friction at the punch-workpiece interface is considered null and the reduction in height applied is 5%.

Cases of Study

In this study the geometry of the workpiece is changed from one to another case. Different values of the geometrical parameters have been chosen for all the cases to be studied. These values are a selection obtained from previous works.

Keeping the height of the billet constant, the width of the billet is incrementally increased from B to 11B, whereas the height of the billet changes from B to 5B. The values of the geometrical parameters that have been chosen in this study are shown in Table 1.

<table>
<thead>
<tr>
<th>Height ($h$)</th>
<th>Width ($w$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1B</td>
<td>B</td>
</tr>
<tr>
<td>2B</td>
<td>B</td>
</tr>
<tr>
<td>3B</td>
<td>B</td>
</tr>
<tr>
<td>4B</td>
<td>B</td>
</tr>
<tr>
<td>5B</td>
<td>B</td>
</tr>
</tbody>
</table>

As stated by Marin et al. (2008a), the dimensions of the workpiece depend on the size of punch ($B$). In this work, it has been chosen a punch where $B$ is given the value 0,01 m.

FINITE ELEMENT MODEL

Finite Element Software

This study has been developed using the Finite Element Method (FEM) (Rowe et al. 1991). A general purpose software of implicit methodology (ABAQUS/Standard) (Hibbitt et al. 2007) has been used.

Design of the model

The punch has been designed as a rigid part and the workpiece has been modeled as a deformable body. The interaction between a deformable body and rigid body has been defined. The surface of the deformable body is considered as the master surface, while the slave surface is considered a rigid part (Bigot et al. 2008). The contact between the workpiece and the punch has been defined as a mechanical contact property: tangential and normal behavior have been specified. Figure 3 shows the model of one particular case, where $w = 11B$ and $h = 5B$.

![Figure 2: Model studied](image)
Mesh of the billet

The type of element is CPE4R and consists of a continuous, plain strain, linear interpolation and reduced integration element. The billet has been modeled as the aluminum alloy is described below. The size of the elements in the mesh has been chosen proportionally to the height of the billet. Figure 3 shows the mesh of previous case.

![Figure 3: Example of mesh for w=11B and h=5B](image)

Material

The billet has been modeled with an aluminum alloy. The type of material is considered as a strain hardened one. A linear behaviour of strain hardening is assumed for the material. Common mechanical properties of the material are illustrated in Table 2.

Table 2: Mechanical properties of the material

<table>
<thead>
<tr>
<th>Type</th>
<th>E (Pa)</th>
<th>ν</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic</td>
<td>$7 \cdot 10^{11}$</td>
<td>0.33</td>
</tr>
<tr>
<td>Plastic</td>
<td>$Y$ (Pa)</td>
<td>ε</td>
</tr>
<tr>
<td></td>
<td>$7 \cdot 10^8$</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>$3.7 \cdot 10^9$</td>
<td>0</td>
</tr>
</tbody>
</table>

Friction model

A Coulomb friction model is assumed by the Finite Element Software. As a first approach, a frictionless problem has been assumed for all the cases, simulating a well lubricated process.

Output variables

This program allows to ask for several output variables to be analyzed. In the present work two output variables are required: CPRESS and PEEQ, that is, contact pressures and equivalent plastic strains, respectively.

RESULTS

In this work the profiles of contact pressures have been obtained such as significant results. The contact pressures have been calculated for the different cases presented above.
In each graph it is shown that from a given height the profiles of contact pressures are stabilized. In other words, become constant. This phenomenon starts at one particular height. It happens firstly in curves where the widths are lower.

As it is shown in Figure 4 this phenomenon begins in the case of height 3B, where the profile of contact pressures for the width 3B remains constant. When the height of the workpiece increases, the same behaviour is observed. Thus, the same happens with the height 4B but the phenomenon begins in width 5B and so on.

As stated before, ABAQUS is used such as the Finite Element software. Several parameters related to geometrical dimensions, boundary conditions and behaviour of the material, among others, are introduced into the program. When the finite element analysis is complete and the problem converges, the simulation shows how the workpiece is formed and strained permanently under the force of the punch.

In Figure 4 the results of the contact pressures obtained for all the presented cases are shown. In this figure there are several graphs. Each graph represents the results for different widths of the workpiece keeping constant the height.

In these graphs, the true distance of contact on the punch-workpiece interface is represented in x-axis, whereas contact pressures are specified in y-axis. Punch-workpiece interface is the area where the contact takes place and on the rest of the surface contact pressures are null. Only the values of contact pressures for the punch-workpiece interface have been represented.

As it can be observed in the figure, the higher the height of workpiece, the higher the contact pressures obtained. If each graph is analyzed individually (different curves for a given height) also the contact pressures are increased with the width of the workpiece. Hence, the higher both the width and the height, the higher the contact pressures obtained. This behaviour means that the punch suffer greater wear with increasing the size of the workpiece.

When comparing the profiles of the contact pressures in all the cases studied, it seems to be that they have a similar behaviour. The maximum pressure is reached at a point near to the end of the punch while the minimum pressure point is obtained in 0.002 m.

Initially, for the lowest height (B), a smooth curve is shown with a single peak where the maximum is reached. But as the height increases another peak appears. This peak grows with the increase in height. For the case with height 5B, this chip and the point of maximum pressure have the same value.
contact pressures are also achieved in the same area, therefore the maximum wear of the punch will occur in this area. Therefore, considering the indentation process as a manufacturing process and being known the geometry of the workpiece to manufacture then it may be known better the behavior of the punch against the wear.

In future works indentation process will be discussed under other conditions. It will be taken into account the friction between the billet and the tool as friction and lubrication affect the material flow and the deformation characteristics of the workpiece. On the other hand, taking into account where the maximum contact pressures and maximum strains are produced, different geometries of the punch should be considered.

REFERENCES


AN ACTIVITY- AND ACTOR-ORIENTED APPROACH FOR THE PROJECT ENGINEERING OF COMPLEX, WEAKLY STRUCTURED PROJECTS

Sven Tackenberg, Sönke Duckwitz, Selvinaz Karahancer, Wolfgang Kabuß, Bernhard Kausch, Christopher M. Schlick

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

KEYWORDS
Project Engineering, Simulation Model, Project Planning, Project Controlling

ABSTRACT
The complexity of a single project – resulting from the amount of activities to be accounted for as well as the required resources – and the amount of simultaneously running projects within an enterprise are rapidly increasing. In order to stay competitive, those factors relevant for the success of project planning and execution, e.g., the duration or the budget of a project, must be specifically identified and realized. On this basis, the Institute for Industrial Engineering and Ergonomics at RWTH Aachen University developed an integrative simulation model that connects the advantages of existing activity- and actor-oriented models. The integrative model presented here is a key component of the Project Engineering approach and enables the automatic creation and prospective benchmarking of complex, detailed project plans. The specific aspects of the new, innovative simulation model were developed in cooperation with enterprises of the process engineering industry and were verified in different development projects.

INTRODUCTION
The planning of simultaneously running projects, which also exhibit a great degree of freedom regarding their execution, results in a highly complex planning problem for the project planners. Thus, project planners face challenges that cannot be solved without a scientifically proven planning method and the application of software tools. Existing tools for project planning are often very process-oriented and do not sufficiently account for the behavioral aspects of the workers. Furthermore, it must also be noted that the development of the methods and software tools does not take into account the work organizational perspective of a project.

PROJECT ENGINEERING
The aim of the new Project Engineering Approach is the creation, analysis and optimization of effective and efficient project plans through methods of engineering science (Schlick et al. 2008; Tackenberg et al. 2008). For this purpose activities of planning are necessary which involve decision making on the basis of a systematic decision preparation in order to identify future events (Eversheim and Schuh 1996). In the context of a decision preparation, the project’s work systems – i.e., the cooperation of one or more workers with the work equipment – are substantiated in order to fulfill the predetermined conditions. This paper therefore aims to present an approach that enables project planners to model, simulate and prospectively evaluate varyingly complex project constellations, including the typical uncertainties and interactions within weakly structured projects. More specifically, the following questions shall be answered with the model:

- Which tasks or which activities must be executed in order to achieve the project goal?
- Which predecessor/successor relationships or degrees of freedom exist in the execution of activities?
- Which workers and work equipment lead to the effective and efficient execution of the task?
- What effects do uncertainties in the planning phase have on the course of the project?

These questions characterize different project-specific dimensions. The structure of a project refers to the chronological sequence in which the tasks must be processed. Here, a distinction between deterministic and stochastic sequences can be made. Within the deterministic order, the relationship between different activities is clearly identified through technical or organizational conditions – e.g., activity A must be finished before activity B can be launched, or A and B must take place simultaneously. However, if degrees of freedom exist regarding the execution of tasks, the structure of the project can, dependent on the planning target variables, be varied either manually by the project planner or automatically through the simulation model. Furthermore, there are different possible combinations of workers and work equipment, so-called project configurations for the execution of a task. The identification of permitted and, with regard to the planning targets, effective and efficient project configurations for the task to be executed, is a central aspect of the Project Engineering Approach. The allocation of workers and work equipment is hereby significantly determined by project structure and the available capacity (workers, work equipment, project budget, etc.). The performance of a project is the result of the characteristics and interactions between the project structure
and the project configurations. The project performance characterizes the specific progression for a considered project scenario, i.e., concrete values of planning goals, tasks, resources, etc. Important pieces of information are: specific points in time, effort and characteristics of an activity for solving a project-specific task as well as the occurrence of events that are afflicted by a probability.

Attention should be paid to the fact that the determination of the project structure and the project configurations, in order to generate the desired project performance depend on the planner’s targets, does not only occur before the start of a project. Instead, nowadays there are demands for an integrated approach for project planning, controlling and optimization during the duration of a project. Thus, the goal is to achieve versatile projects whose structure, configurations and performance can be easily adjusted to unconsidered events. As a result, the project plan can be interpreted as a cycle that can be passed through iteratively and that is re-initiated when events, such as a disruption, occur. The process of project planning takes place in four phases. Each phase is linked to an essential question to be answered by the project planners (Fig. 1):

- **Analysis**: What is the initial situation of the project? First, it is necessary to find out at which stage the project is. The result of the analysis is a characterization of existing opportunities and shortcomings in the project.

- **Goal alignment**: Which goals exist for the individual project or for the entirety of the projects in the field of investigation? The focus is thereby on the definition, including the loading, of project and organization-specific specifications for an individual project. The goals can vary during the course of a project; thus, being behind schedule with a project can lead to modified goal values (increase in project duration, decrease in project resources) of the considered projects.

- **Project planning**: What do the structure and the configurations of the project look like? Here, the development of detailed project plans, i.e., the concrete description of possible ways to achieve the project goals, takes place. The generated project plans offer action options for the realization of a desired project behavior.

- **Project optimization**: What does an effective and efficient project plan for the predetermined goals look like? The focus is on the systematic identification and consequent implementation of potentials of improvement regarding the generated project plans. In particular, the aim is to create agility in the project, i.e., the capability for rapid changes and adjustments to the present challenges.

Project planning and execution, as introduced above, must therefore be understood as a continuous process with a large amount of cycles. Project planning simulation models and tools can support the planner in all four phases of the presented cycle. Below, existing simulation approaches will be classified. This enables us to classify the newly developed integrative object for project simulation and optimization into the field of research.

![Figure 1: Four phases of project planning and controlling](image)

**STATE OF RESEARCH – SIMULATION**

Simulation of development projects as a field of research is very heterogeneous. There are, for example, several approaches to characterize the structure of a project (Cho et al. 2001; Kao et al. 2002; Zülch et al. 2004). A good table to classify simulation approaches from an industrial engineering aspect is presented by the VDI 3633, part 6 (2001). Regarding the VDI 3633, the main point is the worker’s degree of personal action within a work system. According to this guideline, the simulation models can be divided into activity-oriented and actor-oriented approaches. Both areas can be further differentiated in regard to the involved person's degree of inclusion in the model: activity-centered, personnel-integrated, and person-centered (Tab. 1).

<table>
<thead>
<tr>
<th>Activity-oriented</th>
<th>Actor-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>In activity-oriented simulation models the model dynamics are determined by the activities.</td>
<td>In actor-oriented simulation models the model dynamics are determined by the persons.</td>
</tr>
<tr>
<td><strong>Activity-centered</strong></td>
<td><strong>Person-centered</strong></td>
</tr>
<tr>
<td>Persons are not explicitly taken into account.</td>
<td>Persons are considered only in an undifferentiated manner.</td>
</tr>
<tr>
<td>Personnel-integrated</td>
<td>Persons are considered as trivial resource, an activity can only be processed if the necessary persons are available.</td>
</tr>
<tr>
<td></td>
<td>Persons are considered in the form of queuing models and the model includes basic selection procedures for competing activities.</td>
</tr>
<tr>
<td>Person-centered</td>
<td>Activity processing is controlled by the persons, persons can make autonomous decisions on the basis of behavioural model and properties and skills of the persons are explicitly considered.</td>
</tr>
</tbody>
</table>

The focus of the present paper is the industrial engineering aspect of approaches for the simulation of projects. Hence, the object of investigation consists of person-centered approaches that are neither activity-oriented nor actor-
oriented. Examples for the realization of activity-oriented approaches in which activities used by resources determine the system behavior can be found in Raupach (1999), Huberman and Wilkinson (2005) and Cho and Eppinger (2005). However, actor-oriented approaches in which actors – the participating persons or organizations – determine the system performance with the tasks specified for them are investigated by Levitt et al. (1999), Adler et al. (1995) and Steidel (1994).

A combination of the perspectives, the activity-oriented and an actor-oriented, or even the combination of the main aspects is not currently known by us. Accordingly, the presented simulation model for analyzing projects must be characterized as person-centered and activity-oriented as well as person-centered and actor-oriented.

INTEGRATIVE APPROACH

The integrative approach is introduced in this chapter. This approach combines the advantages of the activity-oriented and the actor-oriented simulation models in one comprehensive solution. In order to enable a systematic consideration of aspects relevant to the mapping of concurrent development projects, the description of the developed meta-model, based on the work system model of Schlick (2007), is structured in the following partial models:

- Partial model of the project organization
- Partial model of the project structure
- Partial model of the work equipment
- Partial model of the worker
- Partial model of the work object

The partial models will not be merged into one consistent total model. Instead, the developed meta-model (Fig. 2) will be systematically separated back into its fundamental elements, the partial models.

Meta-Model

Partial Model of the Project Organization

An enterprise, $E$, consists of the union of numerous organizational units, $OU$:

$$E = OU_1 \cup OU_2 \cup OU_3 \cup \ldots \cup OU_n.$$ 

Each organizational unit, $OU_i$, consists of $n$ workers with $n \geq 1$, where the sum of all workers that belong to an organizational unit $i$ are labeled $AP$:

$$OU_i = \{ AP_{i,1}, AP_{i,2}, \ldots, AP_{i,n} \}$$

Each worker is clearly assigned to one organizational unit; i.e., the organizational units $OU_i$ and $OU_j$ do not have any mutual workers:

$$AP_i \cap AP_j = \emptyset.$$ 

An exception is the explicit release of a worker by the project planner for tasks ($TA$) in other organizational units. For example, this is the case in the creation of autarkic project teams.

All tasks within the projects to be planned (scheduling problem) are clearly assigned to the enterprise’s organizational units.

$$TA(OU_i) \cap TA(OU_j) = \emptyset.$$ 

Partial model of the Project Structure

The focus of the partial model of the project structure is the task that becomes an activity by being processed by workers and work equipment. An activity is an active model element and represents a work step toward the achievement of the project goal set by the planner. Activity $A$ is characterized by:

- $A_i$ = Activity $i$
- $S_{A_i}$ = Starting time of activity $i$
- $E_{A_i}$ = Ending time of activity $i$
- $I_{A_i}$ = Input of activity $A_i$
- $O_{A_i}$ = Output of activity $A_i$

![Figure 2: Meta-Model of the integrative approach (based on UML notation)](image-url)
Each task has at least one task-specific requirement or one specific worker assigned to it. Additionally, each activity can possess a deadline or organizational, project or employee-specific importance.

The correlations between activities indicate a logical and a temporal dependency. Therefore, an activity, $A_{i+1}$, can only be carried out if the predecessor activity $A_i$ was sufficiently processed. The relationships that determine the processing sequence of tasks thereby exhibit both deterministic as well as stochastic portions. Four types of correlations between activities can be differentiated in a development project: serial, parallel, overlapping and iterative.

Activities that have a predefined order, and in which a task must be completed prior to the start of a subsequent one, are summarized by the term serial:

- **$A_i$ before $A_j$**: $E_{A_i} < S_{A_j}$
  
  Activity $A_j$ cannot start until $A_i$ has ended.

- **$A_i$ meets $A_j$**: $E_{A_i} = S_{A_j}$
  
  Activity $A_j$ starts directly after the completion of $A_i$, meaning the ending time of $A_i$ and the starting time of $A_j$ are identical.

Tasks that do not have a predecessor-successor relationship are classified in overlapping or parallel categories:

- **$A_i$ equals $A_j$**: $(S_{A_i} = S_{A_j}) \quad \text{and} \quad (E_{A_i} = E_{A_j})$
  
  Activities $A_i$ and $A_j$ start at the same time and end at the same time.

- **$A_i$ starts $A_j$**: $(S_{A_i} = S_{A_j})$
  
  Activities $A_i$ and $A_j$ start at the same time.

- **$A_i$ finishes $A_j$**: $(E_{A_i} = E_{A_j})$
  
  Activities $A_i$ and $A_j$ end at the same time.

- **$A_i$ overlaps $A_j$**: $S_{A_i} < S_{A_j} < E_{A_j}$
  
  Activity $A_i$ starts after the beginning but before the end of activity $A_j$.

- **$A_i$ during $A_j$**: $S_{A_i} < S_{A_j} < E_{A_j} < E_{A_i}$
  
  Activity $A_i$ is carried out during the processing of $A_j$.

Iteration is a process of repeated activity execution within a project. Two types of iteration exist in the integrative model. These iterations are characterized by operators: “feedback iteration” and “interaction”. A “feedback iteration” describes the relationship between three or more activities with the following features:

- $O_{A_i} \subseteq I_{A_\mu}, \quad O_{A_\mu} \subseteq I_{A_{\mu+1}}, \ldots, \quad O_{A_{\mu+1}} \subseteq I_{A_\mu}$
- Optional $O_{A_i} \subseteq I_{A_i}$

The successful execution of activity $A_\mu$ prompts the feedback operator to start activity $A_\mu$ anew or to end the feedback operation. If an event, in the scope of activity $A_\mu$, causes a review of the output of activity $A_{\mu+1}$, this leads to a feedback decision and thereby to a new initiation of the processing of activity $A_\mu$.

“Bidirectional information exchange or interaction” describes the relationship between two activities ($A$ and $B$), with the following characteristics:

- $O_A \subseteq I_B$ and $O_B \subseteq I_A$ and
- $E_A \leq S_B$.

The completion of activity $A$ prompts the operator to start the execution of activity $B$. Similarly, the operator prompts activity $A$ to start again following the completion of activity $B$. This procedure is carried out iteratively until the result of activity $B$ equates to the project goals.

The characterization of the project structure – branching and merging of parallel task chains – occurs through so-called control joints. The following are distinguished: Starting joints $K_s$, Ending joints $K_e$, Branching (AND-branching $K_{anvOrd}$ and OR-branching $K_{orOrd}$) and linking joints (AND-linking $K_{anvOrd}$ and OR-linking $K_{orOrd}$), Parallelization $K_{par}$ and Synchronization joints $K_{syn}$.

$$K = \{K_s, K_e, K_{anvOrd}, K_{orOrd}, K_{anvOrd}, K_{orOrd}, K_{par}, K_{syn}\}$$

**Partial Model of the Worker**

In the present integrative model, a worker is characterized by:

$$AP = \{AP_Q, AP_K, AP_M, AP_L, AP_e, AP_{BS}, AP_{Status}\}$$

with:

- **$AP_Q$** = Formal worker qualification, $AP$,
- **$AP_K$** = Existing competence,
- **$AP_M$** = Motivation,
- **$AP_L$** = Learning curve,
- **$AP_e$** = Level of familiarization in a specific activity,
- **$AP_{BS}$** = Preferred processing strategy (serial, complete processing or parallel, iterative processing of tasks),
- **$AP_{Status}$** = Worker status, where $AP_{Status} = \{AP_{Ord}, AP_{Work}\}$ is dependent on whether the worker is currently processing an activity ($Status_{AP_{Ord}}$) or if he/she is organizing the pool for “incomplete activities” and providing the activities with priorities ($Status_{AP_{Work}}$). Furthermore, each worker has exactly one task pool.

**Partial Model of the Work Equipment**

One or more pieces of work equipment, $AM$, can be assigned to a task to be used in processing: IT applications and machines, or hand tools. The work equipment needed for the execution of an activity are designated with the necessary capacity and functionality. Additionally, each piece of work equipment is explicitly assigned to an organizational unit (OU).

**Partial Model of the Work Object**

Each activity possesses at least one input object and produces at least one output object. These input and output objects represent work objects, $AO$, and are substantiated in the classes “information group, material object, paper information object, verbal information object and IT information object”. Workpieces, paper information objects and IT information objects are saved, respectively, in material storage, paper storage and in IT applications.

Material storage $= \bigcup_{i=1}^n Workpiece object i$. 
Paper storage = $\bigcup_{i=1}^{n}$ Paper information object i,

IT application = $\bigcup_{i=1}^{n}$ IT information object i.

The output objects contained in the project characterize the project’s progress, i.e., they represent the project’s value-added process.

Simulation-Model

The key question that arises during the development of a simulation model for the new, integrative approach is: How can the planning quality and efficiency in complex, weakly structured development projects be increased through a simulation model that also allows the mapping of humans as an active human model? In particular, this model of a worker can process several tasks simultaneously. Also this model can independently decide which tasks have a higher priority in terms of processing. Therefore, the goal is a model that allows the simulation-supported generation and optimization of projects plans with consideration of the predefined constraints and the bounded rational decision making by the workers during the selection of the tasks.

The activity-oriented (Kausch et al. 2005; Tackenberg et al. 2008) and actor-oriented (Licht et al. 2006; Duckwitz et al. 2007) simulation tools developed at the Institute of Industrial Engineering and Ergonomics form the foundation for the integrative model. The modeling of at least one development project forms the starting point of the simulation-based optimization. For this purpose, the project planner models the project structure using a suitable specification technique, e.g., the C3 method (Kausch et al. 2005), OMEGA (Gausemeier & Fink 1999), etc. Analogous to the activity-oriented approach, during the activity execution sequence, fixed relationships are only given where these are truly necessary. Moreover, the project planner specifies the project configurations in a provided project editor, i.e., the workers and work equipments are specified and the individual organizational units (departments) are assigned. As a result of this assignment, each work space has a worker and a resource pool available for the processing of tasks. To comply with the demands made of a simulation model for the support of the operative multi-project planning, the project planner can, at any time, define further project structures and/or project resources and then integrate these into the planning pool for projects still to be planned. The information contained in the planning pool is shown in an abbreviated form in Figure 3 via a class diagram (also see the chapter above). Very significant is the formulation of the relationships between activities (execution sequence) and the variable allocation of workers and work equipment to tasks under consideration of relevant restrictions regarding the execution of an activity (assignment to a work space, formal professional qualification, competences, etc.) – analogous to the activity-oriented, person-centered approach.

The interface between the activity-oriented, person-centered approach and the actor-oriented, person-centered approach form the assignment of tasks to workers. For the purpose of integration, the similarity between the two approaches was analyzed. It was thus shown that, in particular, organizational units to which the workers are assigned, represent central linking elements. All tasks, that is, the resulting activities of a project within the planning pool, are assigned to the relevant organizational units. If a planner places a new project in the planning pool or if an existing project is modified, the tasks are determined and then stored in the respective organization-specific task pool (organizational task pool).

All tasks have an “inactive” status following the first initialization of all organizational task pools prior to the start of the initial simulation run. Furthermore, the model distinguishes the task status:

- **Task inactive**: This status indicates that, with the initial activation, a task cannot yet be processed. Two significant causes must be distinguished:
  - **Predecessor condition**: An activity cannot be executed if, for a defined sequence relationship, all relevant predecessor activities relevant to the particular activity have not been sufficiently processed, i.e., if the relevant information or work objects are not made available by the other activities.
  - **Waiting (inactive-waiting)**: The necessary workers and equipment are not available for the initial processing of the particular activity or a necessary event (e.g., a date) is still pending.

- **Activity active**: A task becomes an activity that is processed for a specific period of time in a work system through the use of workers, work equipment and work objects when the task switches to the “active” status.

- **Activity interrupted**: An incomplete activity switches from the active status to the interrupted status when at least one worker or piece of work equipment ends the processing. Without a new distribution of the work contents the further continuation of the execution cannot occur.

- **Activity processed**: An activity switches from “active” to “processed” when the remaining effort necessary for the completion of a task reaches 0%. If iterations are required, i.e., if rework for a previously completed task is needed, then the percentage for the remaining effort increases and the task switches into the “interrupted” or “active” status.

The transitions between the individual statuses, just like the processing sequence of the individual tasks, are briefly presented in the following section via an algorithm (Fig. 4). The starting point of a simulation run is the identification of the activities in the individual organization-specific task pools in which the predecessor activities were adequately processed – status “inactive-waiting”. Tasks that have this status can be processed by any worker of the considered organizational unit that has fulfilled the minimum requirements for task execution. The task assignment among the suitable worker results purely stochastically or through the use of functions, which can be a fundamental component of optimization procedures. Once a task is assigned to the worker through the simulation, the task in need of processing then shows up in the worker’s task pool. As a result, during the course of a project, and particularly for several concurrent projects, a person-specific task pool can
contain numerous tasks with varying processing statuses. For example, the model takes into consideration that a worker does not fully complete a task that has already been started, but rather interrupts the processing of the task depending on events or processed time units, then also starting or continuing the processing of a different task. The processing of tasks is implemented as the execution of an activity, whereby the assumption is made in the model that only one activity per worker can be processed in the simulation at a specific point in time. If several tasks in a person-specific task pool are ready for processing, the worker must organize these tasks independently. An algorithm is therefore used that assigns priorities to the tasks within a person-specific task pool, thus determining the processing sequence of the tasks. Hence, two different activity statuses for the task pool of a worker can be distinguished:

- Task unprocessed or interrupted: An unprocessed task was not yet processed by the worker (effort = 100\%), while an interrupted task (effort < 100\%) is not currently being processed due to the prioritization.

- Task active: The task is currently being processed by at least one other worker.

The algorithm implemented for the selection of the tasks presents important decision-making processes for the workers during the prioritization of pending tasks. This algorithm takes into account that a worker compares both the utility as well as the costs of different action alternatives when making decisions for actions (Licht et al. 2006; Schlick and Licht 2005). However, a worker does not always make rational decisions during the course of a project, especially since the given project plans are generally quite abstract and, due to the inherent degrees of freedom, very generic. Workers are prone to seeing short-term tasks as more important than long-term ones due to the operational day-to-day business in an organization. A higher priority results only when the time frame until desired task completion continues to greatly decrease. This behavior is referred to in literature as bounded rational behavior (Kahneman, 2002).

In order to take this behavior into consideration during decision making, the time factor must be included in a prioritization algorithm. The Temporal Motivational Theory (TMT) of Steel and König (2006) manages to do so. The prioritization algorithm of the simulation model presented here is based on the findings of TMT. The priority that a person assigns to a task consists of several aspects. First, each task receives a value for “importance” that results from the significance of the task in question for the particular project \( T_p \), the project’s contribution to the organization’s success \( T_o \) and the importance for the worker doing the processing \( T_w \). The importance of the task is represented by the positive effect of a task selection. At the same time, the temporal aspect during the priority calculation is also considered. The positive effect of task preparation is realized when the activity necessary for the solving of the task is carried out by the task’s particular deadline \( T_{Dead} \) at the latest. In addition to the time span until the deadline, the effort still needed for the task must also be considered since the urgency of a task is significantly determined by the task’s already attained degree of processing. The urgency of a task or activity at time \( t \) results from the quotients of the work time that must still be invested and the time remaining until the deadline. The remaining processing time can then be calculated by the expected activity duration, \( T_{exp} \), and \( \delta_t \), the already reached degree of processing (1):

\[
Urgency = \frac{T_{exp}(1-\delta_t)}{1+\Gamma^t(T_{Dead}-t)} \tag{1}
\]

The negative aspect of a task selection exists in the necessary familiarization with the task. This negative utility is determined through a specified familiarization, \( T_{SR} \), and the person’s level of familiarization, \( \delta_{SR} \). The level of familiarization increases during the processing of activities and then decreases again during the processing breaks. This is because familiarization with the task is once again necessary for the processing of a new activity, depending on
the length of the processing break. Therefore, the priority of a task in this context can be calculated as follows (2):

\[
\text{Priority} = \frac{T_{\text{req}} \cdot (1 - \delta_t) \cdot (1 - \delta_\nu) \cdot K_\nu + I \cdot K_\nu + I \cdot K_\nu + I \cdot K_\nu}{I \cdot (1 - \delta_\gamma) \cdot T_{\text{req}} \cdot K_\gamma}
\]

The individual weighting of the single factors is described by the values \(K_\nu\), \(K_\gamma\), \(K_\nu\), and \(K_\gamma\). \(\gamma\) represents the weighting of a positive yields through the worker and \(\nu\) represents the weighting of the portion of costs. The person-specific definition of decision preferences is made possible through the individual weighting factors of each of the subsaspects contained in the priority calculation.

**Figure 4: Algorithm – Simulation-Model**

Using this algorithm during a simulation run, the tasks currently within the task pool of the particular worker are prioritized in regular intervals or following an event (e.g., arrival of a new task). The task with the highest priority is selected for the worker to process as long as the prerequisites in regard to the availability of work objects, work equipment and necessary workers are fulfilled. Attention must be paid to the fact that tasks in a worker’s task pool have an “active” status in the organizational task pool even if the person in charge of the tasks is not currently processing them. An activity receives the “interrupted” status only when the task shows a processing outlay and all person-specific task pools must be dipped into for the sake of a new allocation. This can be the case, for example, for iterations or a targeted redistribution of tasks upon the addition of another project. Once tasks are completely processed they are taken out of the person-specific task pool. The current task in the organizational task pool is then also marked as “processed”.

**SIMULATION TOOL**

An object-oriented simulation approach was designed and implemented at the Institute of Industrial Engineering and Ergonomics to illustrate the key aspects and specific characteristics of work systems in an activity and actor-oriented simulation model in the way they are used in development projects of the process industry and the automobile industry. The simulation model has a modular structure to fulfill the prospective requirements of the integrated approach. The structure of the simulation model is illustrated as an UML-based component diagram in Figure 5. A simulation environment in Java (JDK 1.6) was implemented as a prototype for the initial simulation studies of process engineering projects.

**Figure 5: Component-Diagram**

The component Simulation serves the initialization of the simulation model for one or multiple concurrent projects, i.e., the component is needed for the definition of the relevant simulation scenarios to be investigated. For this purpose, a XML-file is loaded by the simulation tool. By invoking the component Simulation, the information of the XML-file is transferred to objects of the class K3-NET. If, in the context of multi-project management, interrelations between simultaneously running projects must be analyzed, an appropriate number of objects of the class K3-NET will be generated. An object of the class K3-NET describes the project structure on various levels of aggregation, i.e., the characteristics of activity types – Single-Activity, Sub-Net, Blob, Synchronous Communication (Kausch et al. 2005) – as well as the relation to one another. Every activity, information element, worker, work equipment and work object is represented by a specific distinctive object of the simulation environment. Thereby, the objects derived from the classes “WORKER”, “WORK EQUIPMENT” and “WORK OBJECT” are stored in lists within the so-called “pools”. The information regarding the project structure, including the degrees of freedom during the execution of tasks and the resulting generated activities, are saved locally in the activity objects of the component Simulator. These objects are then selected by a function of the Activity Processor component during a simulation run.
The component *Net Processor* gives access to the *Activity Processor* and verifies the feasibility of the individual project structures. The project structure is termed “net” (main or sub net), and if a net is executable, the simulation run will be launched. For this purpose, the *Activity Processor* selects the tasks according to their deterministic and stochastic sequences of execution, which are defined in the semi-formal project models. For the considered tasks, the assignment of workers and work equipment that satisfy all the requirements are sent to the components *Actor Selector* and *Tool Selector*. Furthermore, the processing of a task is launched and controlled by the *Activity Processor* and additional information regarding the specific characteristic of the resulting activity is reported to the *Activity Listener* for the subsequent evaluation.

If necessary, a data parser can be used for the initialization of the project model in order to achieve a high flexibility in the generation of semi-formal project models. The Data Parser component forms the interface between the simulation environment and the Graphical User Interface (GUI) for the project planners. The integration of a parser enables targeted selection and preparation of the input information. In this manner, the integration of process modeling tools, such as a C3 editor, WOMS (Schneider & Gerhards 2003) or innovative AR solutions for participative process modeling can be implemented without modification to the simulation environment.

**CASE STUDY**

The application of the Project Engineering method on a priori and a posteriori modeled and simulated development projects in enterprises of the chemical engineering industry confirms the validity of the approach. The example project, presented below, is an a posteriori modeled project for the development of a large scale chemical engineering plant. This project was participatively modeled in cooperation with the responsible project planners and leaders and then simulated. The object-oriented simulation tool – without the prioritization algorithm – served as the simulation tool. Overall, the project consisted of 62 activities whose execution sequences were weakly structured and with which enterprise-internal and external organizations were assigned. In the framework of the implemented Project Engineering, the focus was on the analysis of correlations between workers and work equipment as well as on the duration of the project, its costs and the utilization of workforce. Therefore, several critical success factors of influence of the project were varied:

- Number of persons involved in the project
- Qualification of the staff
- Number of available work equipment
- Reliability of the effort estimate

**Results of the Simulation**

The amount of staff employed was varied systematically, starting with the smallest possible number of workers (three) necessary for the successful processing of the project. The expansion of the project team was carried out via the stepwise addition of workers with different qualification profiles. Through the exclusive variation of these influence variables in the first phase of simulation, the changes in the project runs could be clearly attributed to the project’s personnel configuration. In a second phase, the work equipment (software licenses, laboratory equipment, etc.) necessary for project processing were varied in regard to their number and functionality. The focus here was on the one-per-unit stepwise increase of the availability of individual selected work equipment. Furthermore, uncertainties in the experienced-based estimations of activity efforts were taken into consideration by the project planner. In the multiple execution of a simulation run per simulation scenario (specific worker and work equipment endowments), these uncertainties resulted in the variance of the total project duration. Figure 6 presents the total means in dependence on different project configurations.

![Figure 6: Predicted project durations dependent on project configurations](image)

For the present project, the reduction of the project duration is based in particular on the number of workers involved. However, the number of workers is not the only determining factor. Thus, for this development project, the influence of worker-specific, formal qualifications – process engineer (VT) and facility technician (AT) – on the project duration were investigated. The analysis of the simulation data provided evidence that the qualification of the fourth or fifth worker directly influences the project duration. The involvement of a sixth worker, on the other hand, results in an increase in the project duration, which, through a detailed analysis of the project runs, can be explained by higher coordination efforts and an unsuitable qualification profile of the worker.

The interpretation of the simulation results shows that, with a constant basic endowment of 11 tools, the project duration can be lowered by 16% (from 2978 Time Units [TU] to 2486 TU) exclusively through the involvement of additional workers (with the characteristics of Worker 4 - VT, Worker 5 - AT). In addition, if the number of available pieces of work equipment is increased, an reduction of 9% (2243 TU) can be achieved compared to the basic endowment (11 tools). Overall, based on simulation, a
reduction of the original project duration from 2978 TU to 2234 TU, approximately 24%, can be achieved.

CRITICAL REVIEW AND OUTLOOK

Project planners and managers of concurrent development projects are, through this innovative approach, able to analyze the correlations between the success-critical, project-performance determining influence factors. Additionally, by means of project scenarios, they are also able to investigate the effects that organizational changes have. The new object-oriented simulation tool, compared to the previous action-oriented, person-centered Petri-Net based approach, (cf. Kausch et al. 2005; Tackenberg et al. 2007), is especially advantageous regarding the system performance – i.e., the time needed for the execution of one simulation run – as well as regarding the involvement of the workers’ behavior models. It must be noted that the formation of the work group and the correlation with various characteristics of material project resources has a strong influence on the project performance. Therefore the presented simulation model can support project planners in the systematic configuration, analysis and evaluation of influencing variables and in the derivation of organizational variants. Moreover, the use of the simulation tool among industrial partners showed that an increase in a development project’s complexity results in an increasing portion of the potential for project duration reduction remains unused. The relationship between the complexity of a project and the optimization potential, as well as the relationship between the complexity and suitable optimization methods, is still largely uninvestigated, thus forming the theme for future research work.

Acknowledgement

The research was funded by the German Research Foundation according to the transfer project TB61 and to Grant SCHL 1805/3-1.

REFERENCES


AUTHOR BIOGRAPHY

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.

SÖNKE DUCKWITZ received the Dipl.-Ing. and the Dipl.-Wirt.Ing. degrees from RWTH Aachen University, Germany, in 2003 and 2006, respectively. He is currently a research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University. His research interests include modeling and simulation of cooperative product development processes.

SELVINAZ KARAHANCER studies computer science at RWTH Aachen University and works on her diploma thesis at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University.

WOLFGANG KABUSS completed an apprenticeship as mathematical technician assistant. Since September 2007 he is a full employee mathematical technician assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University.

BERNHARD KAUSCH studied mechanical 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.

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 RWTH Aachen University in 1999, and the Habilitation degree (Dr.-Ing. habil.) also in Mechanical Engineering from RWTH Aachen University 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. 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 RWTH Aachen University.
ACTOR-ORIENTED, PERSON-CENTERED SIMULATION OF PRODUCT DEVELOPMENT PROJECTS

Sönke Duckwitz
Torsten Licht
Peter Schnitz
Christopher M. Schlick
Chair and Institute of Industrial Engineering and Ergonomics
RWTH Aachen University
D-52062, Aachen, Germany
E-mail: s.duckwitz@iaw.rwth-aachen.de

KEYWORDS
Simulation models, human decision-making behavior, person-centered, actor-oriented, new product development, project engineering

ABSTRACT
The simulation of product development projects currently occurs almost entirely in a task and process-based manner, while the workers involved are considered in an undifferentiated manner. Many realized simulation models are used for the optimization of project progressions. These models allow the determination of the optimal number of workers to employ, activity sequences or the employment of resources. This article presents a model for the simulation of product development projects in which the workers, and thereby the actual actor in reality, are the focal point. The study takes into consideration the scope of decision making for the workers involved in the project and, through the actions of the workers, generates the model dynamics. A verification study serves for the depiction of the central model components.

INTRODUCTION
Globalization and technological advances pose novel challenges for the developers of new products. In the face of international competition, high-quality products must come into the market at a competitive price in increasingly shorter cycles (Burghardt 2000; Mihm 2003). In order to cope with these market behaviors, the development projects must be processed in parallel since the follow-up project must already be started during the final phase of a current project (Hirzel 2006) and because developers often work on several projects simultaneously. If numerous projects are executed in parallel within an organization, these projects compete for work equipment and workers. Inadequate management of complex product development projects leads to longer project durations, exceeding of the project budget, and can also lead to great follow-up costs in the form of contract penalties and customer and employee dissatisfaction (Hberman and Wilkinson 2005; Mihm 2003). The simulation of the progression of a development project creates the possibility of reaching a greater degree of planning accuracy in early stages of project execution. In addition, it also enables a better estimate of the effects that additional projects have on an organization’s project environment. Simulations during the execution of a project aid in the early detection of weak spots and the evaluation and need-based implementation of counter measures. Therefore, the simulation of development projects provides effective support for project management.

CURRENT STATE OF RESEARCH
In most literary sources of simulation-based planning of work processes, the technical system and the pure process view are the focus of consideration. Humans are thereby regarded as passive resources. Zülch detached himself from this perspective and placed work organizational aspects in the center of observation (Zülch et al. 2002; 2004a; 2004b). The focus is hereby on staff planning, task structuring or human errors. Zülch’s research studies led to VDI guideline 3633, part 6 (2001). According to this guideline, the simulation models can be divided into activity-oriented and actor-oriented approaches. Both areas can be further differentiated in regard to the involved person’s degree of inclusion (see Table 1).

<table>
<thead>
<tr>
<th>Activity-oriented</th>
<th>Actor-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>In activity oriented simulation models the model dynamics are determined by the activities.</td>
<td>In actor oriented simulation models the model dynamics are determined by the persons.</td>
</tr>
<tr>
<td>Activity-centered</td>
<td>Persons are not explicitly taken into account.</td>
</tr>
<tr>
<td>Personnel-integrated</td>
<td>Persons are considered as a trivial resource; an activity can only be processed if the necessary persons are available.</td>
</tr>
<tr>
<td>Person-centered</td>
<td>Persons are considered in more detail. Aspects of human behavior such as human errors and skills are considered in processing activities.</td>
</tr>
</tbody>
</table>

Here, three levels of detail are suitable. In the activity-centered view, persons are mostly undifferentiated or not at all considered. In the personnel-integrated view, persons are...
simply seen as resources that must be available so that tasks can be processed. In the person-centered view, however, persons are viewed as significantly differentiated and their abilities, mistakes and scope for decision making are taken into consideration. Examples for the realization of activity-oriented approaches, in which activities used by resources determine the system behavior, can be found in Raupach (1999), Huberman & Wilkinson (2005) and Cho & Eppinger (2005). However, actor-oriented approaches in which actors (the participating persons or organizations) determine the system behavior with the tasks specified for them are investigated by Levitt et al. (1999), Adler et al. (1995) and Steidel (1994).

Despite the possibilities of activity-oriented simulation models, projects and particularly development projects often progress differently than planned. Development times are regularly exceeded and the project budget is not adhered to (Mihm 2003). The workers involved in the project are a significant reason for these delays because they do not always behave in the ways that the project planner predicted. This behavior is referred to as bounded rational behavior (Kahneman 2002). The workers within product development especially have a very large scope for decision making in regard to their task execution. The scope for decision making is hereby not limited to just the contextual design of the tasks, but also to the temporal choice of execution time spans. The simulation of product development projects, with particular consideration of the scopes for decision making, and thereby a focused direction of the simulation dynamics toward the human characteristics and behavior, presents a very interesting approach from a work organization perspective.

PERSON-CENTERED SIMULATION MODEL

An actor-oriented and person-centered simulation approach was developed at the Institute for Industrial Engineering and Ergonomics at RWTH Aachen University. This approach focuses on the decision-making behavior of the workers in regard to the processing sequence of the tasks. The most important aspects of a product development project are hereby taken into consideration within different partial models (see Fig. 1).

Partial Model of Products to be Developed

The product to be developed is described in this partial model in relation to its components and the interactions between the components.

Partial Model of Organizational Structure

The partial model of organizational structure reflects the position of the workers in the organizational hierarchy. Here, the leading position can be differentiated from the administrative position. Every worker has a specific rank within the organizational structure that represents his/her position within the organization.

Partial Model of Operational Structure

The partial model of operational structure reproduces – where absolutely necessary – the predecessor and successor relationships of activities and their temporal overlap. These relationships are taken into consideration by the persons during the activity selection process.

Partial Model of Worker

The partial model of the worker involved presents the central model component. The self-organization of the worker (who has a large scope for decision making) is contained in this partial model. This model is responsible for the selection of activities and thereby determines the entire dynamics of the simulation system. The organizational and decision-making behavior of the worker in this person-centered simulation model is examined more closely below.

Partial Model of Work Equipment

The partial model of work equipment provides resources necessary for the processing of activities. Hence, activities can only be processed if the necessary work equipment is available in the form of machines or special workplaces. A concentration on relevant work equipment in order to limit the parameterization effort should be strived for. Relevant work equipment is that which is limited in quantity and can therefore lead to bottlenecks in project execution or work equipment that, when considering the “costs” goal criterion, has a significant influence on the project costs due to its high costs, and that then must be taken into consideration in the simulation for the evaluation of different project scenarios. A further prerequisite for the processing of an activity can be the availability of necessary workers. Thus, during the building of a prototype, a developer may require the support of additional workers. Since these workers often have considerably less decision-making leeway in the selection of their activities, they do not require individual decision-making functionality during the simulation. Therefore, they are integrated into the partial model of work equipment.

SELF-ORGANIZATION OF THE WORKER

A worker within the simulation model has either the “organization” or the “processing” status. The “organization” status can be divided into three subphases:
• “Activity selection”: the prioritization and thereby the decision-making behavior of the workers
• “Active negotiation”: the initiation of a negotiation at the start of a team activity
• “Passive negotiation”: the reaction to requests regarding the start of a team activity

An overview of the “activity selection” phase is given in Fig. 2.

Figure 2: “Organization” status, “Activity selection” phase and “Processing” status

At the beginning of an activity selection phase, each worker searches through the global task pool for new tasks that were assigned to him/her. It must hereby be noted, however, that the assignment of persons to activities does not occur in the simulation itself. Rather, it takes place during the parameterization performed by the project planner. If all activities are carried over into the person’s individual task pool, then the activities are prioritized with the help of the algorithm which is displayed in detail below. As soon as an activity has a positive priority – a negative priority indicates an activity that, e. g. due to a lack in preparatory work, cannot yet be executed – the activity with the highest priority is selected and the respective work equipment is reserved. Then, if the work equipment is available, the activity is selected for processing. Next, it is checked whether the activity is one that can be processed by individual workers or if it is a team activity that requires several workers. A meeting, in which numerous people must be present at the same time, can be such an activity. If a team activity is the focus, the person switches into the “active negotiation” organization phase. Alternatively, the person can directly begin processing, thus also switching into the “processing” status.

The “processing” status is also depicted in Fig. 2. The activity’s degree of completion is increased in each time unit during processing. This occurs until the activity has been entirely processed, in which case the person switches back to the “organization” status, or until one personally freely definable time span, the so-called organization cycle time, has run out and the person halts processing to then initiate a new decision for an activity, thereby again switching to the “organization” status and “activity selection” phase.

Aside from these two fundamental aspects, the persons’ negotiations also play an important role in the worker submodel. Development projects are largely carried out in teams and numerous activities exist that cannot be processed by one person alone, instead requiring synchronous processing by several persons. Meetings in particular play a role here, yet so does the integration of several subcomponents of a product, the construction of a prototype, etc. The active and passive negotiations are displayed in Fig. 3.

Figure 3: “Organization” status, Active and Passive negotiations by the person

A person that selects a team activity after prioritization takes on the active part of the negotiation. This person then determines the additional team members and sends requests to all team members. Subsequently, this person waits for the return of responses from the team members, evaluates these responses in regard to the overall activity utility and then either rejects the team activity and returns to the “activity selection” or sends the requests from the beginning of the team activity to all the team members and then waits for the start of the team activity, which is linked to a switch to the “processing” status. Passive negotiation (see Fig. 3) is performed by each person following every action that takes up simulation time. Each
person checks if new requests for a team activity exist and then answers these. Next, it is checked whether or not a request for the start of a team activity exists and if the starting time of this activity was already reached. If this is the case, the person’s current action is interrupted and he/she switches to the “processing” status. The team activity is then simultaneously processed by all team members. If no queries or requests exist at the start of a team activity, or if the starting time of a team activity has not yet been reached, the person continues performing his/her previous action.

DECISION-MAKING OF THE WORKERS

The focus of the model is the algorithm that provides the activities with priorities, thereby determining the sequence of activity processing. This algorithm characterizes many of the developer’s important decision processes during the activity selection.

When making decisions, a person compares the gains with the costs of all alternative actions (Schlick and Licht 2005). However, humans do not always make rational decisions in accordance with a predetermined project plan. Instead, humans are much more likely to see short-term activities as more important than long-term ones. Only a cumulative reduction in the remaining processing time of an activity leads to the assignment of a higher priority. This behavior is referred to as bounded rational behavior (Kahneman 2002). In order to take this behavior into consideration during decision making, the time factor must be included in a prioritization algorithm. Many decision and motivation theories exist that investigate and display the different aspects of human decision making. However, these theories that stem from various scientific disciplines only present a certain amount of application-specific aspects of decision making. The Temporal Motivational Theory (TMT) of Steel and König (2006) comprises key insights of motivation theories from different scientific disciplines and thereby forms a wide-spanning theoretical framework for human negotiation and decision-making behavior. The theory thereby takes into account fundamental insights of picoeconomics, Expectancy Theory, Cumulative Prospect Theory and Need Theory, delivering clues about human behavior in decision-making situations such as financial decision-making, governmental or customer behavior (Steel and König 2006).

The TMT describes the utility of a decision as the sum of all positive and negative possible results. In addition, it takes into consideration different weighting possibilities for the advantages and disadvantages. As a result, the individual risk preference can then be depicted. Furthermore, the temporal aspect of the win or loss occurrence is also considered. This is shown in the TMT formula:

\[
Utility = \sum_{i=1}^{\text{results}} \frac{E^+_i \cdot V^+_i}{Z + \Gamma^+ (T - t)} + \sum_{i=k+1}^{n} \frac{E^-_i \cdot V^-_i}{Z + \Gamma^- (T - t)}
\]

where:
- \( \text{results} \) = Positive/negative results
- \( E \) = Probability of occurrence of the result \( i \)
- \( V \) = Value of result
- \( \Gamma \) = Loading factor
- \( T - t \) = Time delay of result occurrence
- \( k \) = Number of possible positive results

\( n \) = Number of all possible results
\( Z \) = Constant

The utility of a decision is the result of the sum of all possible positive and negative results, each weighted with an individual \( \Gamma \) factor. It is then evident that, across the \( (T-t) \) factor, the remaining time until the result occurrence has an influence on the utility. Therefore, an event that occurs early on receives a higher utility value than an event that has a higher expected value but that occurs at a later point in the future. The display of bounded rational behavior of persons is thus possible through TMT.

Prioritization Algorithm

The prioritization algorithm of the simulation model presented here is based on the findings of Temporal Motivational Theory. The priority of an activity is hereby composed of numerous aspects.

First, the positive results of a decision are examined. Each activity has a certain importance, both for the executing person (\( I_p \)) as well as for the organization (\( I_c \)), which can be supplemented by the importance of the project (\( I_p \)) within the organization’s entire project environment. The importance of the activity presents the positive effect of an activity selection. In addition to this, the temporal aspect is taken into consideration on the positive side of the activity prioritization. The positive effect of the processing of an activity occurs at the latest when the activity must be completed, i.e., when the activity’s deadline, \( T_{\text{Dead}} \), has been reached. Then, the effort that must still be invested in an activity prior to the deadline must also be taken into account since an activity’s urgency is significantly determined by an activity’s achieved degree of completion. The urgency \( u(t) \) of an activity at time \( t \) is the result of the quotient of the work time that must still be invested and the respective time left until the deadline. The remaining processing time can then be calculated by the expected activity duration, \( T_{\text{exp}} \), and the already achieved degree of completion, \( \delta \):

\[
u(t) = \frac{T_{\text{exp}}(1 - \delta)}{1 + \Gamma^{-}(T_{\text{Dead}} - t)}
\]

Overall, the activity selection’s positive aspects result in the following formula:

\[
\text{pos. Utility} = \frac{T_{\text{exp}}(1 - \delta_c)(I_p \cdot K_p + I_c \cdot K_c + I_a \cdot K_a)}{1 + \Gamma^{-}(T_{\text{Dead}} - t)}
\]

\( K_p \), \( K_c \), and \( K_a \) are represented by the individual worker through personal loading factors of the individual importance aspects.

The negative aspects of an activity selection have numerous causes. First of all, familiarization with the activity is necessary. This cost factor is determined in each activity through the specified set-up time, \( T_{\text{ST}} \), and the person’s degree of familiarization, \( \delta_{\text{ST}} \). The degree of familiarization increases during the processing of activities and then decreases again during the processing breaks. This is because familiarization
with the task is once again necessary for the processing of a new activity, depending on the length of the processing break.

Aside from the familiarization effort, the information deficits can also negatively influence the activity selection. It must therefore also be considered whether all information necessary for the processing of an activity was already generated in the preceding activities. In order to determine these information deficits, the activity information that must be generated for other activities must first be defined. In addition, it must be defined to what extent the activities must be processed so that the necessary information output can be transferred to successive activities. This information is presented in the operational structure submodel. A matrix is hereby specified that then defines the particular work progress necessary in predecessor activities, as shown by the example in Fig. 4.

![Information dependency matrix](image)

Figure 4: Information dependency matrix

The matrix presented in Fig. 4 indicates that Activity 1 (A1) does not require any information from Activities 2 and 3 (A2, A3), while A2 requires information from A1 that is then available when the degree of completion of A1 is 50%. A3 requires information from A1 and A2. This information is available as soon as the degree of completion of A1 attains 100% and the degree of completion of A2 attains 30%. The (standardized) information content of the activities can be given on the main diagonal of this matrix in order to provide the activities with different levels of importance regarding their information production for successor activities. The information content is identical for all activities in the example presented here. The information deficit for each of the activities is calculated with the necessary degree of completion of the individual activities and the degree of completion (δ) already achieved. In the initial situation of the project shown in Fig. 4 the degree of completion for all activities is 0%.

The information deficit of Aᵢ, Dᵢₜᵢᵢ, caused by Aᵢ, is calculated by using the following formula:

\[
Dᵢₜᵢᵢ = \begin{cases} 0 & aᵢ - δᵢ < 0 \\ (aᵢ - δᵢ) \cdot aᵢ, & \text{otherwise} \end{cases}
\]

The overall result for the information deficit of Aᵢ is:

\[
Dᵢₜᵢ = \sum_{j \in i} Dᵢₜᵢᵢ \cdot aᵢj
\]

A1 thus has an information deficit of 0 since preparatory work is not needed for this activity. A2, however, has an information deficit of 0.5 and A3 has a total information deficit of 1.3.

This information deficit – provided with a worker’s personal weighting factor – becomes part of the negative aspects of activity prioritization.

A further negative aspect of the activity selection is found in the potential competence deficits of an activity’s executor. Similar to the determination of the information deficit, a definition of the required competencies for each activity in the operational structure submodel is necessary. The competencies of each person are stored in the partial model of the worker so that the degree of fulfillment of the competencies of each activity via each worker can be calculated. Competence deficits (Dᵢₜᵢᵢ) also enter the prioritization algorithm with a personal weighting factor for the individual workers.

The negative aspects of a decision for an activity always occur at the activity selection point in time. Therefore, from the cost perspective, the consideration of temporal aspects can be foregone (Tᵢᵢ = 0). Hence, the negative aspects of an activity selection result in the following formula:

\[
\text{neg. Utility} = \frac{1}{\Gamma} \cdot (cᵢT · (1 - δᵢ) · TᵢT · KᵢT + cᵢP · KᵢP · Dᵢ + cᵢE · KᵢE · Dᵢ)
\]

For the priority \(p_i\) of an activity relevant to this context, the following is the overall result:

\[
p_i = \text{pos. Utility} - \text{neg. Utility}
\]

The definition of different decision preferences of workers involved in the project is facilitated through the individual loading factors of the separate subaspects (Ki) within the priority calculation. The ci factors serve the formula setting. Using this algorithm, the project activities are prioritized in regular intervals and the activity with the highest priority is then processed as long as the prerequisites in regard to work equipment availability and the participation of the necessary team members are fulfilled. The algorithm thereby presents important aspects that determine the bounded rational behavior of humans (according to Steel and Königs 2006):

- A short deadline leads to a higher priority.
- Higher importance leads to a higher priority.
- The utility of an activity’s processing decreases during processing.
- Opportunity costs (familiarization efforts) are taken into consideration.
- Activities whose information input was already generated are processed preferentially.
- Activities whose competence profile is fulfilled by the person are preferred.
- Inter-individual differences in the processing strategy exist.

**Negotiation of a Team Task**

During the negotiation regarding the selection of a team task, it is no longer just the individual priority of the specific worker that must be considered. Instead, the overall team must be regarded. A team activity is only then carried out
when the utility of the team task outweighs its cost for the entire team. It must hereby be noted that the utility and cost aspects do not have the same definition as in Section “Prioritization Algorithm”.

In order to guarantee an overall examination, it is necessary to identify the utility and the costs of each individual team member. This is ensured through the requests and their responses. The person that initiates the negotiation directs requests to all team members and then receives responses from them in which the workers indicate their individual utility, their individual costs and the earliest starting point of the team activity. The utility of a team activity is, just as for the individual activities, inherent in the individual priority assigned to the activity by each team member. The costs are incurred by the fact that, at the start of a team activity, each person must interrupt another activity or cannot begin a new one. The priority of this activity then represents the costs of the selection of a team activity.

However, it is not just the individual priority assignment that is decisive during the selection of a team activity. The positions of the individual persons within the organizational hierarchy are also crucial. Thus, a manager’s desire to accomplish a team activity represents a greater weight than when only persons of the same hierarchical level are involved. The individual utilities and costs of the team activity are also weighted either strongly or weakly according to the rank – defined in the partial model of organizational structure – of the particular person. The sum of the weighted utilities and the negative costs are then examined. If this sum is greater than zero, the overall team utility outweighs the overall team costs and the team members are prompted to begin the team task at the earliest possible starting point.

**SIMULATION MODEL.**

Although the previous model (Licht 2008) was implemented purely as a Petri Net model, the simulation model presented in this study was realized through the combination of components of Petri Net software (Renew) and an object-oriented programming language (JAVA). In particular, for reasons of clarity, and because of the power of the implementation language and environment, but especially the performance (duration of a simulation run), the model’s greatly enhanced functionality and the related higher level of complexity demanded a major departure from Petri Net formalism. The advantages of the Petri Net structure in the form of activity concurrence were gained through the combinatorial approach. The object-oriented direction of the approach provides a good foundation for future expansion.

**VERIFICATION – CASE STUDY**

The verification of the simulation model presented here will occur by means of a case study that contains the most important model elements. For this purpose, the development project depicted in Fig. 5, which consists of ten activities and three participating developers, is defined. The product consists of six components that are integrated into one total product. The project is finalized by a test run of the developed product. For the project execution, the following assumptions regarding the expected duration and deadline of the individual activities are made (see Table 2).

<table>
<thead>
<tr>
<th>Activity</th>
<th>Duration [h]</th>
<th>Deadline [h] (after project start)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>8</td>
<td>500</td>
</tr>
<tr>
<td>A2</td>
<td>40</td>
<td>5500</td>
</tr>
<tr>
<td>A3</td>
<td>40</td>
<td>5500</td>
</tr>
<tr>
<td>A4</td>
<td>30</td>
<td>4500</td>
</tr>
<tr>
<td>A5</td>
<td>30</td>
<td>4500</td>
</tr>
<tr>
<td>A6</td>
<td>80</td>
<td>5000</td>
</tr>
<tr>
<td>A7</td>
<td>20</td>
<td>5500</td>
</tr>
<tr>
<td>A8</td>
<td>16</td>
<td>6500</td>
</tr>
<tr>
<td>A9</td>
<td>16</td>
<td>6500</td>
</tr>
<tr>
<td>A10</td>
<td>16</td>
<td>7500</td>
</tr>
</tbody>
</table>

In regard to the information dependencies, it is assumed that A7 can already start once A4 and A5 each reach a 70% level of completion, that A8 requires 80% of A6 and that A10 can be started only when A8 and A9 are 100% complete. Further information dependencies are shown in Fig. 5.

![Figure 5: Example project](image)

Furthermore, it is assumed that A2 and A3 are identical in regard to their additional factors (importance, set-up time, etc.), just like A4 and A5. However, it is also assumed that Person 2 has a slight competence deficit (80% degree of competence fulfillment) in respect to the processing of A5. An exemplary simulation run is displayed in Fig. 6.
Several key elements of the decision making process are made clear through Fig. 6. A1 is processed first due to the greatest amount of urgency at the start. The processing of A2 and A3 by worker 1 occurs profoundly parallel (see Fig. 6a) since these activities have an identical priority and the processing changes due to the constantly changing higher level of urgency between both of the activities. A4’s processing is initially preferred over that of A5. This can be explained by the influence of worker 2’s competence deficit. However, during the course of the project the processing becomes increasingly parallel because the competence deficit is compensated by the increased urgency of A5 (see Fig. 6b). A7 begins once the demanded degree of completion of A4 and A5 has been nearly reached. Fig. 6b also shows that the parameterization of the required information input is not strict, because the various effects the prioritization algorithm contains suspend each other. This means, e.g., that activities are already started although the information deficit is not resolved. In the exemplary simulation run, the increasing urgency of A7 causes worker 2 to start processing A7 even though A5 did not reach the required degree of completion of 70%. The influence of an information deficit can be observed in Fig. 6c. A8 begins once the demanded degree of completion of A6 has been reached. A9 as a team task begins following the completion of activities A2, A3 and A7 synchronously for workers 1 and 2. The start of A10 does not occur until all other activities have been completed, as defined during the parameterization of the example project.

CRITICAL REVIEW AND OUTLOOK

The actor-oriented, person-centered simulation model presented in this article delivers an approach to the mapping of workers’ human decision-making behavior during the selection of activities. Therefore, numerous important behavior patterns are considered that can be summarized by the bounded rational decision concept. Although a fundamental version of this simulation model was thoroughly verified on the basis of Timed Stochastic Colored Petri Nets (Licht et al. 2006; Licht 2008; Duckwitz et al. 2007) by using development project datas of an enterprise, a verification and sensitivity analysis for the model presented in this study must still be performed. To do so, the use of actual development project data is particularly vital. Furthermore, a validation study using several projects carried out in an organization is also aimed for in order to better identify the practical relevance of these simulation approaches.

Acknowledgements

The research was funded by the German Research Foundation (DFG) according to Grant SCHL 1805/3-1.
REFERENCES


AUTHOR BIOGRAPHY

SÖNKE DUCKWITZ received the Dipl.-Ing. and the Dipl.-Wirt.Ing. degrees from RWTH Aachen University, Germany, in 2003 and 2006, respectively. He is currently a research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University.

TORSTEN LICHT completed an apprenticeship as electronics technician and worked for two years for STN Atlas GmbH, a large German naval technology company. He completed his M.S. degree (Dipl.-Ing.) in Mechanical and Industrial Engineering in 2002 at the University of Applied Sciences Kiel. He worked as a research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University and received his Ph.D. degree (Dr.-Ing.) in Mechanical Engineering from RWTH Aachen University in 2008. He is now working as a project controller at Conergy AG, Hamburg.

PETER SCHMITZ studies computer science at RWTH Aachen University and works as student assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University.

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 RWTH Aachen University in 1999, and the Habilitation degree (Dr.-Ing. habil.) also in Mechanical Engineering from RWTH Aachen University 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. 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 RWTH Aachen University.
THE EFFECT OF THE MODEL COMPOSITION TO THE STRUCTURAL PROPERTIES OF PROCESS 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, Veszpré,
Hungary
E-mail: gerzson@almos.vein.hu

KEYWORDS
process models, model composition, 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 (degree of freedom, structural solvability, model decomposition, dynamic degree of freedom, differential index, e.g.) of these models by using the so-called dynamic representation graph. The structure of the dynamic representation graph is suitable for the determination of the mentioned solvability properties. The most common methods in the modeling practice for the construction of models of complex systems are the union of submodels and hierarchical modeling. Our goal is to investigate the effect of the model union to the solvability properties, especially to the differential index. We show how the representation graph of a complex model can be built up from the representation graphs of submodels. The effect of the structure of submodels and their joining points to the structure of the complex graph and the conclusions drawn from the complex graph structure to the solvability properties are also investigated.

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):

\[ y_i = f_i(x, u), \quad i = 1, \ldots, M \]
\[ u_k = g_k(x, u), \quad k = 1, \ldots, K \]

where \( x \) (\( j = 1, \ldots, N \)) 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 \( X = \{ x_1, \ldots, x_M \} \), \( U = \{ u_1, \ldots, u_K \} \) 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 \( x_1, \ldots, x_n \) 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](image)

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):

- **\( S \)-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 \( \langle S \rangle \) 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 \( \langle S \rangle \) and \( \langle S' \rangle \) are treated the same way during the analysis.

- **\( G \)-type variables**: A variable assigned to a specific value of a left hand side is a \( \langle G \rangle \)-type variable. Unlike the \( \langle S \rangle \)-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 \( \langle S \rangle \) can be calculated by simple substitution into the corresponding equation. These variables become secondarily labelled by type \( \langle S \rangle \), and this process can be repeated if necessary. Omitting all vertices labelled primarily, secondarily, etc. by type \( \langle S \rangle \) 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 \( \langle G \rangle \) 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:

\[
\begin{align*}
  z_1' &= f(z_1, z_2, t), & z_1(t_0) = z_{10} \\
  0 &= g(z_1, z_2, t)
\end{align*}
\]

The most important structural computational property of DAE models is the differential index (Gear and Petzold 1984). 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 \( z_1' \) 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 nonsingular.

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 \(<S\rangle\) or \(<G\rangle\) can also be labelled preliminary, secondarily, tertiary or etc. by type \(<S\rangle\).

2. Non-singularity of \(g_{22}\) results in an underspecified part on the graph. In this part those algebraic variables appear, which cannot be calculated from algebraic equations and those derivative 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.
   - \(l_0\) is the set of the derivative variables belonging to the overspecified subgraph,
   - \(D_0\) is the set of the derivative variables referring to the differential variables of set \(l_0\),
   - \(l_1\) 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 \(l_1\), ...
   - \(l_k\) 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 \(l_k\), ...

2. Let \(n\) be the smallest natural number for which the set \(D_k\) contains some derivative variables of the underspecified subgraph. Then the differential index of the model is

\[ n + 2 \]

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).

**STRUCTURAL ANALYSIS OF SIMPLE MODELS USING THEIR REPRESENTATION GRAPHS**

In this section, simple, small sized, dynamic models are investigated using their representation graphs. We show the influence of the change of the modelling goal (and so the model specification) and the modelling conditions to the differential index. The examples used in this and next sections are based on examples of (Moe 1995).

**Example 1. – Perfectly stirred tank reactor**

Suppose a perfectly stirred tank reactor and let the concentration of its inlet flow be denoted by \(c_0\). The change of concentration in the tank can be described by the following equation:

\[
c' = \frac{q}{V}(c_0 - c)
\]

(3)

where \(c\) is the concentration in the tank, \(q\) is the outlet flow rate and \(V\) is the volume of the tank.

**Case a)** Let us assume that we know the concentration of the inlet flow in the function of time: \(c_0 = c(t)\), and we want to determine the concentration of the outlet flow. The standard form model consists of the following equations:

\[
c = \frac{1}{r}c'
\]

\[
c' = \frac{q}{V}(c_0 - c)
\]

\[
c_0 = c_0(t)
\]

Given: \(c(t_0), c(t)\), \(c_0\);

Constant: \(q, V\);

To be calculated: \(c\) as a function of time.

Since the structural properties of the model described by representation graph can be investigated based on the structure of the static graphs, and these properties are independent from the arcs connecting individual static graphs to each other, we illustrate only one static graph as a representation graph of models for the sake of simplicity.

The representation graph of this simple model is shown in Figure 2.a. The reduced graph is an empty graph in this case indicating the differential index is equal to 1.

We remark that the substitution of the condition \(c_0 = c_0(t)\) into the Equation (3) results in a model of technological system with only one differential equation, so the differential index would be equal to 0.

**Case b)** Let us assume now that the modelling goal is the dynamic design of the same system, i.e. the determination of the necessary inlet flow concentration in order to ensure the required outlet concentration \(c = c(t)\). The standard form model is the following:

\[
c = \frac{1}{r}c'
\]

\[
c' = \frac{q}{V}(c_0 - c)
\]

\[
c = c(t)
\]

Given: \(c(t_0), c(t)\);

Constant: \(q, V\);

To be calculated: \(c_0\) as a function of time.

![Diagram](image_url)

Figure 2: The Representation Graphs of the Example 1.
In this case, there are an underspecified and an overspecified subgraphs on the representation graph (see Figure 2.b) referring to the differential index greater than 1 value. The differential index can be calculated based on the structure of the representation graph:
\[ I_0 = \{ c \} \]
\[ D_0 = \{ c' \} \]
Since the vertex referring to the derivative variable \( c' \) can be found in the underspecified subgraph, therefore \( n = 0 \) and \( v_d = n + 2 = 2 \).

**Example 2.a. - Liquid mixer model**

Suppose a liquid mixer tank having one inlet and one outlet flow (see Figure 3.) The inlet flow consists of two components \( A \) and \( B \). The two components have different density. There is a certain amount of liquid in tank at \( t = t_0 \). The feed is perfectly mixed with the tank liquid. The density of the liquid in tank, the flow rates and the mol fractions of the components are functions of time. The number of moles \( (N_i) \) of components \( A \) and \( B \) can be described by the following equation:
\[ N_i = F_0 x_i - F x_i \]
where \( i = \{ A, B \} \)
where \( F_0 \) and \( F \) are the inlet and outlet flow rate, and \( x_0 \) and \( x_i \) are the mol fraction of the component \( i \) in the inlet and outlet flows, resp. Let \( p_i \) denote the pressure of the liquid at the bottom of the tank, \( a \) the area of the tank and \( M_w \) the molar weight of the component \( i \). The outlet flow rate depends on the liquid pressure \( (p_i) \) and the valve constant \( (k) \).

![Figure 3: Liquid Mixer Tank with Variable Volume](Image)

The modelling goal is to calculate the liquid composition in the tank. The standard form model consists of the following equations:
\[ N_A = \int N_A' \]
\[ N_B = \int N_B' \]
\[ N_A = F_0 x_{A0} - F x_A \]
\[ N_B = F_0 x_{B0} - F x_B \]
\[ N = N_A + N_B \]
\[ x_A = N_A/N \]
\[ x_B = N_B/N \]
\[ M_A = M_{A0} + M_{A0} x_{A0} \]
\[ p_A = p_0 + (M_{A0} N_A)/a \]
\[ F = k(p_0 - p_0)^{1/2} \]
Given:
\[ N(A(t_0), N_B(t_0), x_{A0}, x_{B0}, F_0, V) \]
Constant:
\[ v_A, v_B, \]
To be calculated: \( N_A, N_B, F \) as functions of time.

The representation graph of the model is shown in Figure 4. The reduced graph is an empty graph because there is no implicit equation in the model, therefore the differential index \( (v_d) \) is equal to 1.

![Figure 4: The Representation Graph of the Example 2.a](Image)

**Example 2.b. - Liquid mixer model with constant tank volume**

Suppose a liquid mixer tank as in Example 2.a but let the volume of the liquid in the tank \( V \) be constant in this case (Figure 5.). Let \( v_A \) and \( v_B \) be the molar specific volumes of components \( A \) and \( B \), resp. The other assumptions are the same as in Example 2.a. The modelling goal is to calculate the liquid composition in the tank, again, but the modified volume condition must be taken into account.

![Figure 5: Liquid Mixer Tank with Constant Volume](Image)

The standard form model consists of the following equations:
\[ N_A = \int N_A' \]
\[ N_B = \int N_B' \]
\[ N_A = F_0 x_{A0} - F x_A \]
\[ N_B = F_0 x_{B0} - F x_B \]
\[ N = N_A + N_B \]
\[ x_A = N_A/N \]
\[ x_B = N_B/N \]
\[ v = v_A x_A + v_B x_B \]
\[ V = N v \]
Given:
\[ N_A(t_0), N_B(t_0), x_{A0}, x_{B0}, F_0, V, \]
Constant:
\[ v_A, v_B, \]
To be calculated: \( N_A, N_B, F \) as functions of time.

An overspecified and an underspecified subgraph can be found on the representation graph (see Figure 6.) and the differential index can be determined based on their structures:
\[ I_0 = \{ N_A, N_B \} \]
\[ D_0 = \{ N_A', N_B' \} \]
Since the vertices referring to derivative variables \( N_A', N_B' \) can be found in the underspecified subgraph, therefore \( n = 0 \) and \( v_d = n + 2 = 2 \).
**Example 3.a – Gas pipe model**

Suppose a simple short gas pipe (see Figure 7.). The pipe has a valve at the exit and there is heat transfer with the surroundings through the wall but the heat capacity of the wall can be neglected. The modelling goal is the calculation of the pressure ($p$) and the temperature ($T$) inside the pipe. The following assumptions are taken: ideal gas, turbulent flow, ideal mixing, uniform pressure and temperature inside the pipe. The inlet flow ($F_0$) and inlet temperature ($T_0$) are functions of time. The surrounding temperature and the temperature of the pipe wall are equal and constant, the heat transfer through the wall is denoted by $Q$. The outlet gas flow ($F$) is expressed by a valve equation for sub critical flow and the valve constant is $k$. The volume of pipe ($V$) and heat capacity of the gas ($c_v$) are constant. The internal energy is denoted by $U$, the enthalpy by $h$ and the number of moles by $N$.

![Diagram](image)

**Figure 7: A Short Gas Pipe**

The standard form model consists of the following equations:

\[
\begin{align*}
N' &= \int N \\
U' &= \int U \\
N' &= F_0 - F \\
U' &= F_0 h_0 - F h + Q \\
T &= U(N, c_v) \\
h &= c_v T \\
s &= p V - N R T \\
F &= k (0.5 (p^2 - p_0^2))^{1/2}
\end{align*}
\]

Given: $N(t_0)$, $U(t_0)$, $F_0$, $h_0$, $Q$;

Constant: $R$, $c_v$, $V$;

To be calculated: $p$, $T$, $N$, $U$, $F$ as functions of time.

The representation graph and the reduced graph can be seen in Figure 8. There is a Menger-type complete linking in the reduced graph, therefore the differential index ($\nu_d$) is equal to 1.

![Diagram](image)

**Figure 8: The Representation Graph and the Reduced Graph of the Example 3.a.**

**Example 3.b – Gas pipe model with specified pressure**

Assume the gas pipe model as it was described in previous section but let the pressure of the gas be specified. The other assumptions are the same as earlier. The modeling goal is to calculate the temperature as a function of time and to find the exit stream that satisfies $p = p(t)$.

The standard form model consists of the following equations:

\[
\begin{align*}
N' &= \int N \\
U' &= \int U \\
N' &= F_0 - F \\
U' &= F_0 h_0 - F h + Q \\
T &= U(N, c_v) \\
h &= c_v T \\
s &= p V - N R T \\
F &= k (0.5 (p^2 - p_0^2))^{1/2}
\end{align*}
\]

Given: $N(t_0)$, $U(t_0)$, $F_0$, $h_0$, $Q$, $p(t)$;

Constant: $R$, $c_v$, $V$;

To be calculated: $T$, $N$, $U$, $F$ as functions of time.

In this case, an overspecified and an underspecified subgraph can be found on the representation graph (see Figure 9.) and the differential index can be determined based on their structures:

\[
\begin{align*}
I_0 &= \{ N, U \} \\
D_0 &= \{ N', U' \}
\end{align*}
\]

Since the vertices referring to differential variables $N'$, $U'$ can be found in the underspecified subgraph, $n = 0$ and $\nu_d = n + 2 = 2$. 

78
STRUCTURAL ANALYSIS OF COMPOSITE MODELS

In this section, more complex composite models are built from the simple dynamic models of the previous section. The goal is to investigate the effect of this "build up process" of simple, small sized models, i.e the effect of the model composition to the structural properties of the composite models.

Example 4. – Cascade of perfectly stirred tank reactors

Suppose a system consists of \( k \) perfectly stirred tank reactors. A feed of concentration \( c_0 \) is fed into the first tank. The concentrations in the tanks are described by the following equation:

\[
\frac{d}{dt} c_i = \frac{q}{V_i} (c_{i-1} - c_i), \quad i = 1, 2, \ldots, k
\]

where \( c_i \) is the concentration in the tank \( i \), \( q \) is the flow rates from tank to tank and \( V_i \) is the volume of the tank \( i \).

Two different specifications can be added to these equations according to modelling goal:

a) in dynamic simulation studies the feed concentration \( c_0 \) is given by \( c_0 = c_0(t) \);

b) in dynamic design the product concentration \( c_k \) is given by \( c_k = c_k(t) \).

The representation graphs referring to these specifications can be seen in Figures 10a and b. These graphs can be considered as multiplications of the representation graphs in Figures 2a and b. In the first case, the reduced graph is an empty graph, therefore the differential index is equal to 1. In the second case, there are under and overspecified subgraphs on the representation graph and based on their structures:

\[
I_{k+1} = \{c_k\} \\
D_k = \{c_k'\}
\]

Since the vertex referring to the derivative variable \( c_1' \) can be found in the underspecified subgraph, \( n = k - 1 \) and \( v_d = n + 2 = k + 1 \).

Figure 10: The Representation Graphs of the Example 4.

The effect of the increasing differential index of the cascade model can be followed on the representation graph: the underspecified and the overspecified subgraphs move increasingly further from each other as the cascade elements are inserted. The path between the derivative variable \( c_1' \) of the underspecified subgraph and the differential variable \( c_k \) of the overspecified subgraph is increasingly longer (the direction is not taken into account) and along this path the differential and derivative variables are located alternately.

Example 5. – Sequence of mixing tanks

Suppose that a system consist of a sequence of \( k \) mixer tanks (see Figure 11). Let the volume of liquid in the tank \( j \) be constant while in the other tanks the liquid volumes are variables. The model of the constant volume tank is described in Example 2b, while the models of the other tanks are the same as the model in Example 2a. The following assumptions are held: A liquid feed stream is fed into the first tank. The feed consists of two components \( A \) and \( B \). The liquid flows from the first tank through the system. The other assumptions are the same as in Example 2a and b.

Figure 11: Sequence of liquid tanks
The model of this cascade system using the models of Example 2 is the following:

\[ N_{a_i} = \{ N_{a_i} \} \]
\[ N_{b_i} = \{ N_{b_i} \} \]
\[ N_{a_i} = F_{a_i} x_{a_i-1} - F_{a_i} x_{a_i} \]
\[ N_{b_i} = F_{b_i} x_{b_i-1} - F_{b_i} x_{b_i} \]
\[ N_{a_i} = N_{a_i} + N_{b_i} \]
\[ x_{a_i} = N_{a_i}/N_i \]
\[ x_{b_i} = N_{b_i}/N_i \]
\[ M_{a_i} = M_{a_i} x_{a_i} + M_{b_i} x_{b_i} \]
\[ p_{a_i} = p_{a_i} + (M_{a_i} \cdot N_i) / a \]
\[ F_i = k \left( p_{a_i} - p_{b_i} \right)^{1/2} \]
\[ v_j = v_j x_{a_i} + v_p x_{b_i} \]
\[ V_j = N_j v_j \]

Given:
\[ N_{ad}(t_0), N_{bd}(t_0), i = 1, \ldots, k \]
\[ x_{ad0}, x_{bd0}, F_{a_0}, F_{b_0}, V_{j0} \]

Constant:
\[ M_{a_i}, M_{a_i}, a, k, p_{a_0}, v_{a_i}, v_{b_0} \]

To be calculated: \[ N_{ad}, N_{bd}, F_i \] as functions of time.

This model is built up from \( k-1 \) differential index 1 models and one differential index 2 model according to the liquid mixing system. The representation graph of the whole system can be constructed easily from the representation graphs of the simple models (see Figures 4 and 6). The resulted graph can be seen in Figure 12.

An overspecified and an underspecified subgraph can be found on the representation graph and the differential index can be determined based on their structures:

\[ I_0 = \{ N_{a_i}, N_{b_i} \} \]
\[ D_0 = \{ N_{a_i}', N_{b_i}' \} \]

Since the vertices referring to differential variables \( N_{a_i}' \). \( N_{b_i}' \) can be found in the underspecified subgraph, \( n = 0 \) and \( v_d = n + 2 = 2 \).

In this example the union of the representation graphs of submodels has been created in such a way that the position of the underspecified and overspecified subgraphs referring to the higher differential index in the extended graph is unvaried to their original position, therefore the differential index of the complex model is the same as of the model of Example 3.b.

Example 6. – Long gas pipe model

Suppose a system consists of a long gas pipe as an extension of Example 3 (see Figure 13). The pressure in the pipe is assumed uniform while the temperature varies in the pipe. The pipe is divided into two control volumes, as it can be seen in Figure 13. The gas in the first control volume of the pipe is described by a model similar to the gas pipe model in Example 3.b. The pressure of the second part is treated as a variable, therefore the model of this part is described by a model similar to the low index gas pipe model of Example 3.a. The modeling goal is to calculate the pressure and temperature of the gas in the pipe. The other assumptions relating to the technological system are the same as in case of Example 3.

![Figure 13: Long Gas Pipe](image)

The standard form model consists of the following equations:

**control volume 1**

\[ N_{i1} = \{ N_{i1} \} \]
\[ U_{i1} = \{ U_{i1} \} \]
\[ N_{i1} = F_{i0} - F_{i1} \]
\[ U_{i1} = F_{i0} h_{i1} - F_{i1} h_{i1} + Q_{i1} \]
\[ T_{i1} = U_{i1} / (N_{i1} c_i) \]
\[ h_{i1} = c_i T_{i1} \]
\[ s_{i1} = p_{i1} V_{i1} - N_{i1} R T_{i1} \]
\[ s_{i1} = 0 \]
\[ p_{i1} = p_{i2} \]

**control volume 2**

\[ N_{i2} = \{ N_{i2} \} \]
\[ U_{i2} = \{ U_{i2} \} \]
\[ N_{i2} = F_{i1} - F_{i2} \]
\[ U_{i2} = F_{i1} h_{i1} - F_{i2} h_{i1} + Q_{i2} \]
\[ T_{i2} = U_{i2} / (N_{i2} c_i) \]
\[ h_{i2} = c_i T_{i2} \]
s_2 = p_2 V_2 - N_2 RT_3 \quad s_3 = 0 \\
F_3 = k(0.5 \ (p_2^3 - p_0^3))^{1/2}

Given: N(t_0), N_2(t_0), U_1(t_0), U_2(t_0), F_0, h_0, Q_1, Q_2;
Constant: R, c_p, p_0, V_1, V_2;
To be calculated: T_1, T_2, N_1, N_2, U_1, U_2, F_1, F_2 as functions of time.

The representation graph of the complex model can be built up from the representation graphs of the models in Examples 3.a and 3.b. The resulted graph is shown in Figure 14. According to the model equation p_1 = p_2 there is only one vertex referring to the variables p_1 and p_2.

There is an under and an overspecified subgraph on the representation graph of the complex model, too, which is coming from the representation graph of Example 3.b. The relative positions of the underspecified and the overspecified subgraphs do not change during the union of representation graphs, therefore the differential index is equal to 2, invariably.

Figure 14: The Representation Graph of the Example 6.

**CONCLUSION**

In this paper we investigated the solvability properties of complex dynamic systems when they are built up from simple models. We have shown that the representation graph can be used efficiently for the investigation of the differential index during the model composition process, too. The representation graph of the complex model can be easily built up from the representation graphs of the simple models according to the linking of the technological subsystems. If one of the subsystems has greater than one differential index then the under and overspecified subgraphs referring to this higher index can be found in the representation graph of the complex model, too. The change in the relative position of the underspecified and the overspecified subgraphs has an effect to the value of differential index. If these subgraphs move further from their original positions the value of the differential index increases as it can be seen in Example 4.b. If their relative positions do not change during the built up process then the value of the differential index of the complex system is equal to the value of the differential index of the subsystem having the higher value as it is shown in Examples 5 and 6.

**REFERENCES**


**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 Loránd University, Budapest. She is currently associate professor in the Department of Mathematics and Computing at the University of Pannonia, Veszprém, Hungary, where she lectures on linear algebra, operation research, theory of algorithms and complexity.

**MIKLÓS GERZON** 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 he lectures on control theory, measurement theory and technique.
COMPUTER SIMULATION OF CROSS-SECTIONAL VARIATION FOR WORSTED WOOL TYPE YARN

Ionut Neagu  
Pascal Bruniaux  
ENSAIT " L'école Nationale Supérieure des Arts et Industries Textiles "  
2, allée Louise et Victor Champier,  
Roubaix, 59056, France.  
ionut.neagu@ensait.fr  
Pascal.bruniaux@ensait.fr

Dorin Avram  
Neagu Rozenberg Adrian  
Technical university "Gh. Asachi"  
Str. D-trie Mangeron nr. 53, Corp  
TEXI, Iasi-700050, Roumania  
dorin.avram@tuiiasi.ro  
adyneagu@yahoo.com

KEYWORDS  
Fiber-yarn, simulation interfaces, model design, virtual reality, Computer Aided Design.

ABSTRACT  
The paper makes reference to establish yarn’s geometrical structure determined by the cross-sectional variation given by the index of unevenness of yarns finesse, and the fibers finesse, therefore using this simulation to model a woven or hosiery in a 3d graphical environment.

ORIGINALITY, POSITION  
Recent developments of fiber properties have established that the diameter in spinning is overwhelmingly the most important property. Variation in diameter is normally limited with 5% such as the other properties have no observable input in yarn properties, spinning performance, fabric stiffness or next-to-skin comfort [1]. In this paper are presented some tools for computer simulation of cross-sectional variation for worsted wool type yarn, starting from a data set of fiber characteristics. This is an original approach in the problem of computer simulation of cross-sectional variation of yarn’s.

APPROACH

Theoretical consideration regarding the sectional variation

Mathematical modeling of the yarns characteristics

Yarn medium count, m/g, is expressed with:

\[ N_m = \frac{Nm}{n_s} \times Cs \]  \hspace{1cm} (1)

Where:

\[ Nm \] - is the Fiber’s medium count, m/g;
\[ n_s \] - Mean number of the fibers in the yarn section;
\[ Cs \] - yarn’s retraction coefficient

Limits for the coefficient of variation

Limits for the coefficient of variation are determined as number of fibers from the yarn’s cross section variation.

Limits for the coefficient of variation, for the yarns from blends are determined with the expression:

\[ CV_{lim} = \sqrt{\frac{a_1 \times CV_{lim1}^2 + a_2 \times CV_{lim2}^2}{2}} \]  \hspace{1.5cm} (2)

Where: \( CV_{lim1}, CV_{lim2} \) - Limits for the coefficient of variation of yarns obtained separately from the 1st component, respectively from the 2nd component.

Limits for the coefficient of variation are expressed with:

\[ CV_{lim} = \frac{100 \sqrt{\frac{1}{n_s}}}{\%} \]  \hspace{1cm} (3)

Where:

\[ n_s \] - Represent the number of fibers from the yarn’s transversal section for each separate component [1].

Yarn diameter statistical variation is expressed with a normal repartition [2].

Fig. 1. - Yarn diameter variation.

From where:

\[ D_{min} = D - 1.96 \times \sigma \]  \hspace{1cm} (4)
\[ D_{max} = D + 1.96 \times \sigma \]

Where:

\[ D \] - Main diameter of the yarn;
\[ \sigma \] - Standard deviation of the yarn diameter.

Standard deviation is expressed with:

\[ \sigma = \frac{CV_{ef} \times D}{100} \]  \hspace{1cm} (5)

Where:

\[ CV_{ef} \] - effective coefficient of variation, expressed with:

\[ CV_{ef} = CV_{lim} \times I \]  \hspace{1cm} (6)

Where:

\[ I \] - yarn unevenness index.
\[ I = 1.2 - 1.25; \]

The variation diagram of the yarn is obtained by determining the classes’ amplitude, with the expression:
\[ a = \frac{D_{\text{max}} - D_{\text{min}}}{n_c} \]  
(7)

Where: 
\( n_c \) - are the number of the classes (20) and the variation of diameters following a normal function for which the density of repartition is:

\[ f(D_i) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(D_i-D)^2}{2\sigma^2}} \]  
(8)

Normal distribution

The normal distribution (Gauss) is used in this paper for distribution of the experimental measurements [3].

Bezier polynomials

Usage of Bezier polynomials is for approximating the plane curvature implemented to unite the diameters generated by the software. One of the principal problems regarding the modeling of the objects is the design of the structure. In some cases very simple, it is sufficient to use some analytical entities such as: points, segments, circles, conics, etc. For more complex structures, this type of design represents the disadvantage that any change in the object, is almost impossible. From this reason, it is more advantageous to give a few reference points and to generate curve between them, which approximate the object form. Such synthetic entities have Bezier parametric curve’s [4].

Software for simulation of the cross sectional variation

The software is named SIMUDIM and is simulating the variation of cross sectional diameters of the wool type yarns; we used the PERL programming language to construct the software, in the figure 2. is presented the way the program work.

The input data of yarns and fibers characteristics are described in Table 1, and the output in the Table 2.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Input data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiber’s medium count</td>
<td>Nmf</td>
</tr>
<tr>
<td>The number of the fibers from the transversal section of the component 1</td>
<td>Nsc1</td>
</tr>
<tr>
<td>The number of the fibers from the transversal section of the component 2</td>
<td>Nsc2</td>
</tr>
<tr>
<td>Limits for the coefficient of variation of the component 1</td>
<td>CVlimc1</td>
</tr>
<tr>
<td>Limits for the coefficient of variation of the component 2</td>
<td>CVlimc2</td>
</tr>
<tr>
<td>Limits for the coefficient of variation</td>
<td>CVlim</td>
</tr>
<tr>
<td>Effective coefficient variation</td>
<td>CVef</td>
</tr>
<tr>
<td>The minimum diameter of the yarn</td>
<td>Dm</td>
</tr>
<tr>
<td>The maximum diameter of the yarn</td>
<td>DM</td>
</tr>
<tr>
<td>Amplitudes of the classes</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Output data</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of the fibers from the transversal Section of the yarn</td>
<td>Ns</td>
</tr>
<tr>
<td>Yarn’s coefficient of retraction</td>
<td>Cs</td>
</tr>
<tr>
<td>Fiber’s medium count of the component 1</td>
<td>Nmc1</td>
</tr>
<tr>
<td>Fiber’s medium count of the component 2</td>
<td>Nmc2</td>
</tr>
<tr>
<td>Fiber’s medium count</td>
<td>Nmmed</td>
</tr>
<tr>
<td>The percent of the first component</td>
<td>%sc1</td>
</tr>
<tr>
<td>The percent of the second component</td>
<td>%sc2</td>
</tr>
<tr>
<td>Index of unevenness</td>
<td>I</td>
</tr>
<tr>
<td>The mean diameter of the yarn</td>
<td>D</td>
</tr>
<tr>
<td>Number of the experimental diameters</td>
<td>Nrmas</td>
</tr>
</tbody>
</table>

Conclusions

In this paper we proposed to offer an image of simulation and modeling field, introducing the fundamental notions, specifics terminology, and usual applications and to present a series of result of the surveys made by different authors. Even if there are a lot of studies in the field of modeling and simulation of textiles there is no study concerning the dynamic behavior of the yarn, using as inputs the characteristics of fibers, more precisely the modeling of the yarns structure taking into account the fibers characteristics. Using curves, as Bezier polynomials for interpolation, is a new approach, especially as an application for the worsted wool type yarns.

The advantages which the companies could profit from the deployment at an industrial scale of virtual technologies:

• By using the 3D tools, the companies can reduce the hours consumed by the process of development of products. This alternative has like consequence the reduction of the raw material cost and working time.
• Integration of virtual technologies will make possible to a quickly response from companies to produce products by eliminating the need for building the prototypes, the samples of the market and perhaps even
the products themselves before they are sold. This would have as a consequence the reduction of the time needed to launch a product on the market and would allow a process practically integrated of development adaptable to the geographical distances.

REFERENCES


Negoa I, D.Avram, P.Bruniaux, 2006 “The state of using the artificial neuronal networks in forecasting the characteristics of textile fibers”, Master en Roumanie, research in GEMTEX


84
and Exhibition, Simulation in Industry, pages 433-436, Dresden, Germany, 23-26 October.


BEHAVIOUR METHODOLOGY
ON THE USE OF ASSOCIATION RULE AND CLUSTERING IN STUDENT PROGRESSION

Kifaya S.Qaddoum
Faculty of Information Technology
Philadelphia University
Amman, Jordan
E-mail: sweetkq@hotmail.com

KEYWORDS
Association Rule, Classification, Data Mining.

ABSTRACT

This paper idea aims to discuss and evaluate a modeling approach for student progression. It is developed as a component of an adaptive achievement system. At the beginning of the process, associations of student achievement results are found based on each student’s factors that affect learning process, which finds the relationship between student progression during years of study and understanding the modular scheme, that finds the main effect of enrolling on the correct modules i.e. getting the right advice and student support regarding choosing modules, meeting all the necessary prerequisites, having summer courses, and taking in consideration the student's high school grades, as well as finding the relationship between modules type and student gender . Clustering (J.A. Fernández Pierna, 2000), or unsupervised classification, method is employed to model this task.

The goal of clustering is (Fisher, D., 2002)to objectively partition data into homogeneous groups such that the within group object similarity and the between group object dissimilarity are determined. Clustering here is used to model student achievement according to predefined criterion functions that measure similarity among students who grant certain goal having the same conditions using data collected from University Database. A clustering method is developed for this step. We evaluated the student progress according to associations between different factors using data collected. We concluded the performance of those groups using these two approaches.

Now, the need for solid information about student progression and how to improve it has only grown in importance for state policy. The compelling metaphor of increasing flow through the “educational pipeline” is now common in state policy discussions, fueled by more vocal recognition by business and civic leaders of the importance of the critical “supply chain” of educational capital in their states (e.g. Miller and Ewell 2004).

INTRODUCTION

Since it has been introduced, Association Rule Mining (ARM) (Agrawal, et al., 1994), has received a great deal of attention by researchers and practitioners among data mining. ARM is an undirected or unsupervised data mining technique, which works on variable length data, and it produces clear and understandable results. It has a simple problem statement, that is, to discover relationships or correlations in a set of items and consequently find the set of all subsets of items or attributes that frequently occur in many database records or examples, and additionally, to extract the rules telling us
how a subset of items influences the presence of another subset.

ASSOCIATION RULE MINING

The association mining task simply can be stated as follows (Agrawal, et al., 1996): Let \( I \) be a set of items, and \( D \) a database of examples, where each example has a unique identifier (\( iid \)) and contains a set of items. A set of items is also called an itemset. An itemset with \( k \) items is called a \( k \)-itemset. The support of an itemset \( X \), denoted \( \sigma(X) \), is the number of examples in \( D \) where it occurs as a subset. An itemset is frequent or large if its support is more than a user-specified minimum support (\( min \ sup \)) value.

An association rule is an expression \( A \Rightarrow B \), where \( A \) and \( B \) are itemsets. The support of the rule is the joint probability of an example containing both \( A \) and \( B \), and is given as \( \sigma(A \cup B) \). The confidence of the rule is the conditional probability that an example contains \( B \), given that it contains \( A \), and is given as \( \sigma(A \cup B) / \sigma(A) \). A rule is frequent if its support is greater than \( min \ sup \), and it is strong if its confidence is more than a user-specified minimum confidence (\( min \ conf \)).

PROBLEM DEFINITION

The main objective of data mining is to find interesting/useful knowledge for the user, as Rules are an important form of knowledge; some existing research has produced many algorithms for rule mining. These techniques use the whole dataset to mine rules and then filter and/or rank the discovered rules in various ways to help the user identify useful ones.

There are many potential application areas for association rule technology which include catalog design, store layout, customer segmentation, telecommunication alarm diagnosis, and so on.

The data mining task is to generate all association rules in the database, which have a support greater than \( min \ sup \), i.e., the rules are frequent, and which also have confidence greater than \( min \ conf \), i.e., the rules are strong. Here we are interested in rules with a specific item, called the class, as a consequent, i.e., we mine rules of the form \( A \Rightarrow c_i \), where \( c_i \) is a class attribute (\( 1 \leq i \leq k \)).

This task can be broken into two steps:

1. Find all frequent itemsets having minimum support for at least one class \( c_i \). The search space for enumeration of all frequent itemsets is \( 2^m \), which is exponential in \( m \), the number of items.

2. Generate strong rules having minimum confidence, from the frequent itemsets. We generate and test the confidence of all rules of the form \( X \Rightarrow c_i \), where \( X \) is frequent. For example, consider the sales database of a bookstore (Zaki M., 2000) shown in Figure 1, where the objects represent customers and the attributes represent books. The discovered patterns are the set of books most frequently bought together by the customers. An example could be that, "40 percent of the people who buy Jane Austen's Pride and Prejudice also buy Sense and Sensibility". The store can use this knowledge for promotions, shelf placement, etc.

![Figure 1: Mining Frequent Itemset](image)

There are five different items (names of authors the bookstore carries), i.e., \( I = \{ A, C, D, T, W \} \), and the database consists of six customers who bought books by these authors. Figure1 (Liu, B., et al, 2001) shows all the frequent itemsets that are contained in at least three customer transactions, i.e., \( min \ sup = 50 \) percent.

CLUSTERING

Clustering which considered as the most important unsupervised learning problem (Gobert, J., & Buckley, B. C., & Horwitz, P., 2006); so, as every other problem of this kind,
it deals with finding a structure in a collection of unlabeled data. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters. The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. In our case, we are interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection), for students groups according to their progression during the specified time of study.

ASSOCIATION RULE AND CLUSTERING ALGORITHM FOR MODELING STUDENT PROGRESSION

This proposed Algorithm is an iterative algorithm that counts itemsets of a specific length in a given database pass. The process starts by scanning all transactions in the database and computing the frequent items. Next, a set of potentially frequent candidate 2-itemsets is formed from the frequent items. Another database scan is made to obtain their supports. The frequent 2-itemsets are retained for the next pass and the process is repeated until all frequent itemsets have been enumerated.

There are three main steps in the algorithm:

1. Generate candidates of length k from the frequent (k-1) length itemsets, by a self join on \( F_{k-1} \). For example, If

   \[ F_2 = \{AB, AC, AD, AE, BC, BD, BE\} \]

   Then we find that:

   \[ C_3 = \{ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE\} \]

2. Prune any candidate with at least one infrequent subset. As an example, ACD will be pruned since CD is not frequent. After pruning, we get a new set

   \[ C_3 = \{ABC, ABD, ABE\} \]

3. Scan all transactions to obtain candidate supports. The candidates are stored for support counting.

Example Let L3 be \{\{1 2 3\}, \{1 2 4\}, \{1 3 4\}, \{1 3 5\}, \{2 3 4\}\}. After the join step, C4 will be \{\{1 2 3 4\}, \{1 3 4 5\}\}. The prune step will delete the itemset \{1 3 4 5\} because the itemset \{1 4 5\} is not in L3. We will then be left with only \{1 2 3 4\} in C4.

Data Partition: 70% Training, and 30% Validation, since Models are constructed using training data sets and evaluate model performance using validation data sets, and using other data sources as testing data sets.

We used F1 evaluation measure as the base of our comparison, where F1 (Van, R., 1979) is computed based on the following equation:

\[
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}}
\]

<table>
<thead>
<tr>
<th>Table 2 : Data possible sets based on a query in IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Data Retrieved</td>
</tr>
<tr>
<td>Data not Retrieved</td>
</tr>
</tbody>
</table>

Precision and recall are widely used evaluation measures in IR and ML, where according to Table 2,

\[
\text{Precision} = \frac{X}{(X + Y)}
\]

\[
\text{Recall} = \frac{X}{(X + Z)}
\]

To explain precision and recall, let’s say someone has 5 blue and 7 red tickets in a set and he submitted a query to retrieve the blue ones. If he retrieves 6 tickets where 4 of them are blue and 2 that are red, it means that he got 4 out of 5 blue (1 false negative) and 2 red (2 false positives). Based on these results, precision=4/6 (4 blue out of 6 retrieved tickets), and recall= 4/5 (4 blue out of 5 in the initial set).
For objectively partitioning data into homogeneous groups, it is necessary to define criterion functions that measure similarity among objects. Various criterion functions and methodologies have been developed for temporal data clustering systems. They can be grouped into three main categories: (i) proximity based methods, (ii) feature based methods, and (iii) model-based methods.

**EXPERIMENTAL ANALYSIS**

**Data Description**

I collected and stored all student activities in the database. Data collected from Computer Science I (CS-I) students in 2005 was used for this experiment. The collected data contains information from 166 students depending on a student’s performance and the type of student identified by its learned model.

**Experimental Design**

The first experiment compared the student models generated using the classification approach on the static survey data, and those using the clustering approach on the temporal student online data. The classification model learned from the CS-I students in 2004 was applied to students in Spring 2005 after each answered the six learning behavior related questions. Each student was classified into one of three learning categories: Reinforcement type(A), Challenging type(B), and Regular type(C). For the same group of students, the Markov chain based clustering was applied to the temporal lab data deriving a set of classes corresponding to the set of student learning models. Manually analyzing these models leads to a labeling of learning types for these clusters.

In order to compare whether the student categorization derived from the two approaches resemble each other, we compared the category labels assigned to the students. To determine which approach gives a better categorization of the students, I objectively measured the quality of the models derived in terms of the between cluster dissimilarity and within cluster dissimilarity. The derived student learning models are considered of better quality if the models representing different categories are as unique, or as dissimilar to each other as possible. In addition, the student models are considered better quality if students presented by each category are homogeneous in learning style than if there are subgroups following significantly different learning style.

This proves that, after the first level cluster, the students categorized into the same group share very similar behavior pattern/model. They could not be further split into different groups, as in the case of cluster “C2” (distance value 0.0 is put in the table entry), or only relatively similar models could be derived. In the case of the classification approach, since the first level classification did not successfully partition students into homogeneous groups based on their data.

**Conclusion**

This paper showed that using Association Rule and Clustering was effective in finding relations and associations between students raising among given categories. We evaluated the student progress according to associations between different factors using data collected. We concluded the performance of those groups using these two approaches, where we can mine the expected groups for each student. For future work this study should use different categorization algorithms which handle a dynamic and updated data for the students.

**REFERENCES**


Fisher, D., Data Mining Tasks and Methods: Clustering: Conceptual Clustering,
Handbook of Data Mining and Knowledge Discovery, 388-396, 2002
Li, C., A Bayesian Approach to Temporal Data Clustering using the Hidden Markov Model Methodology, PhD thesis, Vanderbilt University, December 2000

BIOGRAPHY

KIFAYA QADDOUM was born in Amman, Jordan and went to the University of Philadelphia, where she studied Computer Science and also obtained her Master degrees in 2008. Since 2003 she works for one of the United Nations Units in Jordan which called UNRWA that serves the Palestinian refugees in Jordan and other countries, she works as Information Technology Instructor at one of UN colleges, in addition to her interest in research work.
Executing UML 2 Diagrams in ActiveCharts - A formal Semantics for the Combination of Behavior Specifications in the UML 2

Jens Kohlmeyer
Institute of Software Engineering and Compiler Construction
Ulm University
89069 Ulm, Germany
E-mail: jens.kohlmeyer@uni-ulm.de

KEYWORDS: UML 2 Diagrams Execution, Modeling, Software Simulation, Formal Semantics, Model Driven Architecture

ABSTRACT

The semantics for the UML 2 is only informally given, although it is important to define a precise and unambiguously semantics for modeling languages used in model driven development. Precise semantics of models is needed for executable models like in the ActiveCharts approach. In this paper we present the extension of the ActiveChartsIDE to UML 2 state diagrams and sequence diagrams. Based on a stated formal semantics for the language units Actions, Activities, State Machines and Interactions we define a formal semantics for the combination of behavior specifications in the UML 2.1.2 using Abstract State Machines. This includes the specification for the UML meta class Behavior and its specializations as well as formal concepts for communication within the UML.

INTRODUCTION

Modern software development processes, like the Model-Driven Architecture initiative (MDA) [15] [14] proposed by the Object Management Group (OMG), focus on models to describe the structure and behavior of a system in analysis and design phases. Models can be reused in the implementation phase by generating code from them. The Unified Modeling Language (UML) is widely used in Model Driven Development (MDD) and is a standard notation for graphical modeling. However, the UML still has no formal semantics, which possibly leads to individual interpretation of the specification by developers or tool implementors. The notation of the UML 2 (syntax) is defined by the meta model, whereas the semantics is defined by text. It is generally accepted, that a formal, precise and unambiguous semantics is needed for a visual modeling language [17] [28].

Our approach ActiveCharts aims at simplifying software development by reusing seamlessly analysis and design artefacts for implementation and a tight integration of models and code. For our implementation and simulation environment ActiveChartsIDE (see [20], [22] and [24]) a precise semantics of the UML 2 behavior specifications like Actions, Activities, State Machines and Interactions is essential. The ActiveChartsIDE is based on an interpreter for UML 2 activity diagrams. The UML 2 meta model provides a basis for this interpreter. We implemented a representation of the meta model in code, therefore. Models can be imported and the behavior of the models and the application can be visualized with the simulation and debugging component. The UML 2 models are integrated into regular code and vice versa. The control flow of the application is modeled with UML 2 activity diagrams which are interpreted by the ActiveChartsIDE. The limitation to UML 2 activity diagrams is a strong restriction, we extend the ActiveChartsIDE to other UML 2 behavior specifications like state charts or sequence diagrams, therefore.

Several approaches deal with the creation of a formal semantics for parts of the UML. In this paper we analyze the idea of coupling different behavior specifications like state charts or activities in one model and define a formal semantics using the formalism of Abstract State Machines (ASM) [4] to use the models for our simulation environment. The UML offers a variety of specification mechanisms like State Machines, Activities or Interactions. This makes it possible to combine several behavior specifications to describe a system. The composition of different behavior specifications is more than viewing the system from different point of views. In figure 1 we show an example of how the composition of different UML 2 behavior specifications can be used to model a system, such as modeling multimodal user interfaces.

The general behavior of the MP3-player is modeled in an activity chart with two actions, the first displaying a welcome message. The second action is a CallBehavior:Action that references a state machine which models the behavior for playing music and volume control. For simplification reasons we omit the actions on the transitions: if the trigger stop is received, an action stop.action is executed on passing the transition, analog
for the other triggers and transitions.

**Figure 1:** MP3-player: simplified example for Combining Behavior Specifications

There are two basic kinds of combining behavior specifications in UML 2:

1. Each Behavior of a BehavioredClassifier is realized with one type of behavior specification

2. Within one Behavior of a BehavioredClassifier different types of behavior specifications are used

These two kinds of combining behavior can appear in hybrid ways, too.

Our work is based on the UML 2.1.2 Superstructure Specification [16] and is embedded in the ActiveCharts approach [20] [22].

The structure of this paper is as follows: in the following section we briefly state the basic concepts of our approach. We then describe a formal semantics for the UML language units at a glance. In the main part we present a formal semantics for the combination of behavior. Then we picture some ambiguities in the semantics of the UML specification and describe the tool support for executing UML 2 models. Finally we compare our approach with related work. Keywords of the UML are written in italics.

**BEHAVIORAL MODELING IN THE UML 2**

The UML Superstructure Specification in the version 2.1.2 consists of different language units to define several views for behavior of a system: Common Behaviors, Actions, Activities, Interactions, State Machines and Use Cases (see figure 2).

**Figure 2:** Language units for behavioral modeling in UML 2

In the language unit Common Behaviors the core concepts for dynamic behavior are defined. Within the subpackage BasicBehaviors a framework for behaviors is defined. The behaviors are specified by the concrete subtypes of the abstract meta class Behavior.

Figure 3 shows a subset of the meta model for Common Behaviors. A Classifier with an assigned Behavior is a BehavioredClassifier. The BehavioredClassifier is the context of the behavior execution.

Behavior is a specification of how its context classifier changes state over time [16, p. 428].

A behavior execution runs within its context.

**Figure 3:** Subset of the UML meta model Common Behaviors

The structure, semantics and the application of a Behavior are dependent on its concrete subtype. Concrete
subtypes of a Behavior are the meta classes Activity, StateMachine and Interaction.
A behavioral base for the UML 2 is given in the language unit Actions. The other behavioral specifications rely on this base, in other words: Activities, StateMachines and Interactions can use Actions to model behavior (for more information please see [5]). A basic meta class for the coupling of behavior within the Actions is the CallBehaviorAction. A CallBehaviorAction can be used to call another Behavior, as shown in the above mentioned example.

ABSTRACT STATE MACHINES

ASMs were introduced by Gurevich [12], based on the formalism of Evolving Algebras. An ASM is a transition system. It consists of a set of rules of the form if condition then updates. updates is a set of functions, which are simultaneously executed when the guard condition is true. A state is given by interpretations of the rules. All applicable updates (update-set) are executed at once, which leads to a new state of the ASM.

An ASM can be used to formally define the behavior of a system. We use ASMs to define the semantics for the combination of Behavior in the UML 2.1.2.

In basic ASMs only one ASM execution exists at a time, since there is only one ASM agent. In our work we use asynchronous multi-agent ASMs, which allow multiple agents running in parallel, each one performing its own rules. These agents communicate by shared functions. For a further introduction to ASMs we refer to [4] and [18].

FORMAL SEMANTICS FOR THE UML 2 LANGUAGE UNITS

In this section we give an introduction to the semantics of the different behavioral specifications in the UML by means of ASMs in a nutshell. For space reasons, we only present basic parts of the macros relevant for the combination of behaviors. The basic idea is to interpret the execution of Behavior with ASMs.

Actions

An Action specifies a transformation on the state of a system. To allow the combination of behaviors, some specific actions are indispensable: For the coupling of behaviors we need the CallBehaviorAction and CallOperationAction, for communication the AcceptEventAction, SendSignalAction and BroadcastSignalAction are required.

In our approach, each action execution is interpreted by a separate ASM agent. The macro CREATEACTERVICEEXECUTION creates a new ASM agent and assigns rules to it, depending on the concrete type of the action. We support the above mentioned Actions. For the complete description of the life cycle of an action execution including macros for creation, enabling and termination we refer to [21].

Activities

A complete ASM semantics for a selected but comprehensive subset of UML 2 Activities is defined in [21]. The creation of a new activity execution is handled by the following rule:

\[
\text{STARTNEWACTIVITYEXECUTION}(...) \equiv \\
\hspace{1cm} ... \\
\hspace{1cm} \text{exec} = \text{new(ActivityExecution)} \\
\hspace{1cm} ... \\
\hspace{1cm} \text{ASM (exec)} := \text{ACTIVITYEXECUTIONCONTROLLER} \\
\hspace{1cm} ...
\]

The rule STARTNEWACTIVITYEXECUTION creates a new agent ActivityExecution and assigns the macro ACTIVITYEXECUTIONCONTROLLER, starting the execution of the controller loop. The rule ACTIVITYEXECUTIONCONTROLLER processes events from the associated event pool. The handling of these events is specified in further ASMs rules, including rules for the computation and execution of transitions, describing the token flow semantics (see [23] for more information).

State Machines

In [1], [2] and [3] an ASM semantics for state machines in the UML 1.x is given. This work has been adapted to version 2.0 of the UML Superstructure Specification, implementing changes in the meta model and adding rules for the execution of transition paths in [8].

During the initialization of a state machine, the rules GENERATECOMPLETIONEVENT and TRANSITION-SELECTION are assigned to the top-level-agent (TLA). They describe the behavior of the agent by processing events from the event pool, similar to the ACTIVITYEXECUTIONCONTROLLER explained in the previous section. The rule TRANSITIONSELECTION chooses a transition to fire and calls the rule STATEMACHINEEXECUTION [1].

\[
\text{STATEMACHINEEXECUTION}(\text{trans}) \equiv \\
\hspace{1cm} \text{if } (\text{kind}(\text{trans}) = \text{internal}) \text{ then } \\
\hspace{2cm} \text{effect}(\text{trans}) \\
\hspace{1cm} \text{else} \\
\hspace{2cm} \text{seq} \\
\hspace{1cm} \text{ExitState( source(\text{trans}) , ToS)}
\]

96
effect(trans)
ENTERSTATE(FromS, target(trans))

This rule executes the chosen transition. The rule EXITSTATE performs the exit-behaviors (exit) of the possibly nested source states and stops internal behaviors (doActivity). EFFECT executes the behavior of the transition (effect) and ENTERSTATE starts the entering behaviors (entry) and the state internal behavior of the target state. A potential doActivity of a state is controlled by a so-called worker agent.

Interactions

In [11] again asynchronous multi-agents ASMs are used to define a formal semantics for UML 2 Interactions. During the initialization with the rule INITSEQUENCECHART executed by the top-level-agent the different communication partners (lifelines) are instantiated and the macro EXECUTEFRAGMENTS is executed. The macro differs between the varied InteractionFragments and starts the corresponding rules, depending on the kind of the fragment. Relevant for the combination of behaviors is an ExecutionEvent within an OccurrenceSpecification: if the corresponding execution of the current fragment is an ActionExecutionSpecification, the rule STARTACTION is called. The rule STARTBEHAVIOR represents the start of a Behavior. It is called if the execution is a BehaviorExecutionSpecification. For detailed information about the formal semantics for UML 2 Interactions we refer the reader to the work of Fuerst [11].

FORMAL SEMANTICS FOR THE COMBINATION OF BEHAVIOR

In UML, each instance of a BehavioralClassifier has an associated behavior execution. We map the different behavior executions to asynchronously executing ASM agents. Each agent has an event queue polled by the behavior execution controller. Signals used for communication are stored in event pools, according to the UML 2.1.2 specification.

At first, the different meta models for the behavioral specifications as described above are translated into ASM domains and functions. The ASM representation of the meta model is the basis for the interpretation with ASMs. We transferred the existing ASM specifications to the version 2.1.2 of the superstructure specification, implementing changes and considering new elements. We exemplify this procedure on the basis of the meta class Behavior.

static domain Behavior ⊆ Class(from Kernel)
static isReentrant: Behavior → Boolean

static context: Behavior → BehavioralClassifier
static specification: Behavior → BehavioralFeature
static ownedParameter: Behavior → Parameter
static redefinedBehavior: Behavior → Parameter
static precondition: Behavior → Constraint
static postcondition: Behavior → Constraint

For space reasons, we can only present the relevant parts of the ASM rules. The complete rules will be presented in a forthcoming paper.

INITIALIZATION

Here we deal with the initialization of the ASMs. We define an initial agent Initialization which executes the macro INITIALIZEBEHAVIOR. This macro is executed first and creates the initial behavior executions for the BehavioralClassifiers.

INITIALIZEBEHAVIOR ≡
forall object with (object ∈ BehavioralObject)
STARTNEWBEHAVIOREXECUTION(object.classOf.
classifierBehavior, object)

For each object with a behavior specification the macro STARTNEWBEHAVIOREXECUTION is called. The parameter classifierBehavior gets the Behavior which specifies the BehavioralClassifier, the second parameter is the context object for the new behavior execution.

STARTNEWBEHAVIOREXECUTION(...) ≡
case behavior of
Activity:
STARTNEWACTIVITYEXECUTION(behavior, object)
StateMachine:
extend Agent with agent
  rule(agent) := INITSTATEMACHINE
Interaction:
extend Agent with agent
  rule(agent) := INITSEQUENCECHART(behavior)

The macro distinguishes between the different subtypes of Behavior specifications and calls macros which initialize the interpretation of the behaviors as described earlier. STARTNEWACTIVITYEXECUTION creates the necessary agents on its own, whereas the macros INITSTATEMACHINE and INITSEQUENCECHART need the top-level-agent, which has to be defined first.

With these rules we implemented the first of the basic kinds for combining behavior specifications (see the introduction). The second kind for the combining of behavior in UML is handled within the next section.

COUPLING BEHAVIORS
In this section we analyze how the different language units can be combined in a single behavior specification. To call a Behavior from an Activity, a CallBehaviorAction is used (see for instance figure 1, where a StateMachine is called). We define the macro ExecuteCallBehaviorActionExecution, which creates among other things a new behavior execution:

\[
\text{ExecuteCallBehaviorActionExecution} \equiv \\
\text{Self.calledExecution} : \text{STARTBehaviorExecutionFromBehavior}(...) \\
\]

The rule STARTBehaviorExecutionFromBehavior creates a new behavior execution based on the type of behavior, similar as shown above. Since we do not allow to call an Interaction from an Activity, we define a new macro, which allows to call a StateMachine or an Activity, only.

\[
\text{STARTBehaviorExecutionFromBehavior} \equiv \\
\text{case behavior of} \\
\text{Activity:} \\
\text{STARTNewActivityExecution} \text{(behavior, object)} \\
\text{StateMachine:} \\
\text{extend Agent with agent} \\
\text{rule(agent)} \equiv \text{INITStateMachine} \\
\text{INITStateMachine} \\
\]

The UML differs between emergent and executing behavior. The latter is the description of a Behavior of an Object, whereas the emergent behavior results from the interaction of participating objects. We do not think it is useful to allow an executing behavior to call an emergent behavior. Therefore we restrict the calling of behaviors from an executing behavior to other executing behaviors. The other way, calling an executing behavior from an emergent one, is allowed, as discussed later in this section.

In the meta model of a StateMachine the abstract meta class Behavior is associated to a State (entry, exit and doActivity) or to a Transition (see [16], p. 525). The state internal behavior is called doActivity, though it is a Behavior. Indeed it is not restricted to Activities. In [1] the rules EXITSTATE, ENTERSTATE and STARTACTIVITY are defined. EXITSTATE leaves the potentially nested states using the macro SEQUENTIALEXIT which calls the function exit(state). EXIT(state) performs the exit-behavior of the committed state. We transfer this function to a macro, which allows to call another behavior.

\[
\text{Exit(state)} \equiv \\
\]

Self.calledExecution := 

\text{STARTBehaviorExecutionFromBehavior}(...) \\
\]

Again we do not allow an Interaction to be called, due to reasons explained above. Unlike Bürger and Dausend we create a new agent for the exit-behavior. The rules for entering a state are similar to the ones described for exiting a state: in [1] an ASM function entry(state) is defined. We change this function to a rule, which calls the rule STARTBehaviorExecutionFromBehavior to start the associated Behavior. To handle the Behavior connected to a Transition, a function effect(transition) is specified. We have to change this function to a rule, again.

\[
\text{Effect(transition)} \equiv \\
\text{Self.calledExecution} := \\
\text{STARTActionExecutionFromBehavior}(...) \\
\]

Please note, that we do not allow to call a Behavior at a Transition at this moment. This is a change to the original UML specification. A transition in a state machine is supposed to be timeless. A Behavior specifies the change of the context object over time. We are currently working on that, right now we restrict the behavior on a Transition to Actions. Moreover, we restrict the possible Actions in the macro STARTActionExecutionFromBehavior and do not allow explicitly CallBehaviorActions and AcceptEventActions, since they are associated with time.

A worker-agent controls the internal Behavior of a State. The rule STARTActivity is called in the macro ENTERSTATE and creates the new worker-agent.

\[
\text{STARTActivity} \equiv \\
\text{extend Agent with agent} \\
\text{rule(agent)} \equiv \text{STARTBehaviorExecutionFromBehavior}(...) \\
\text{worker(state)} \equiv \text{agent} \\
\]

As aforementioned, we do not allow to call an emergent behavior from an executing behavior and restrict the internal behavior to call an Activity or StateMachine.

The other way, calling an executing behavior from an emergent behavior, as it is done in Interactions, is allowed in our approach. We think it is reasonable to call specifying behaviors from an Interaction. The Behavior in the language unit Interactions is connected to a
single Lifeline. It specifies the behavior of a single communication partner, which may have an associated Behavior. In [11] we introduced the macros StartBehavior, StopBehavior, StartAction and StopAction to enable the combination of an emergent behavior with an executing behavior. An ExecutionSpecification can be an ActionExecutionSpecification or a BehaviorExecutionSpecification, as specified in the UML 2 meta model for Interactions (see [16], p. 463).

\[
\text{StartBehavior}(\text{lifeline}) \equiv \\
\quad \text{extend Agent with agent} \\
\text{rule(agent)} : \equiv \text{StartBehaviorExecution-FromBehavior(...)}
\]

The parameter lifeline is needed to get the BehavioralClassifier and to determine the context object for the Behavior. The rule StartAction does basically the same, but starts an Action instead of a Behavior. The macros StopBehavior and StopAction deactivate the according agents and terminate the Behavior or the Action, thereby.

The aforementioned rules implement the second basic kind for combining different behavior specifications. The hybrid ways of the two basic modes are covered with the stated ASM semantics, too.

Semantical Ambiguities in UML 2 Behavior Specification

In this section, we present some ambiguities within the UML specification. We focus on unclarities which come up from the combination of different behavioral specifications.

For the several language units of the UML unclarities and ambiguities have been published. Fecher et al. [9] [10] illustrate semantical problems consisting of ambiguities, inconsistencies, and unnecessarily strong restrictions of UML 2.0 State Machines. In [21] problems and errors in the superstructure specification 2.0 regarding Activities are described. Crane [6] will forward the results from formalizing the UML Actions and Activities to the authors of the specification in order to improve the documents. In [19] problems regarding signal handling and communication and possible solutions in UML 2.0 Activities are explained. In our work we found errors and unclarities in the language units, too. We demonstrate this with some samples, all of which are based on the version 2.1.2 of the UML Superstructure Specification. The problems in the individual language units will be published in a forthcoming paper.

There are some ambiguous or obscure information in the UML specification. For example, in the specification of the CombinedFragment break in the Interactions language unit [16], p. 468, the terms operand and operator are mixed up. Furthermore, there is an unclarity between the meta model and the specification text for the State Machines: in the meta model there is an association from StateMachine to State via submachineState. In the specification text this association is missing. There are some sophisticated problems, too. In Interactions it is inexplicit, what happens if the consider- and ignore-Fragments are nested. We could continue this, however, we do not emphasis on these aspects in this paper. Hence, in the following we focus on problems regarding the combination of behavioral specifications.

As mentioned above, time at transitions in state machines is a problem. We restrict the Behavior at transitions to Actions, as stated earlier. We allow the entry-and exit-behavior of a State to be behaviors like an Activity, since they belong to the state. An Action is a fundamental part of behavior in UML 2. The language units Interactions and State Machines can use actions, as there is an association to the meta model class Action in their meta models, e.g. an ActionExecutionSpecification in Interactions links to Action. In the language unit State Machines only a Behavior can be linked to a Transition or to a State. But, Action is not a specialization of Behavior, it is derived from NamedElement. We do not think it is reasonable to not allow a simple Action like a SendSignalAction at a Transition. This could be handled with an Activity consisting of a single Action, but there is an overhead for starting and stopping the Activity. For a single action execution this overhead is not necessary. The other language units do have Actions explicitly in their meta models.

We identified a problem using a AcceptSignalAction within the Behavior of a Transition. The notation of a Transition can be defined as follows: trigger[guard-constraint]/behavior-expression, each optional.

The behavior-expression is executed if and when the transition fires. [16, p. 574].

An AcceptSignalAction will cause a trigger, and – if the guard is true – the transition will fire and the Behavior is executed. If an AcceptSignalAction is used in this Behavior – which is syntactically correct – the semantics of the Transition is changed: instead of causing a trigger, the AcceptSignalAction waits until the signal is fired and blocks the transition behavior. We do not allow AcceptSignalActions in a transition behavior, therefore. The abstract meta class Behavior has the attribute isReentrant. If this attribute is true, the behavior can be
invoked while it is still executing from a previous invocation. If the behavior is not reentrant, there is only one invocation at a time. This may result in a problem at transitions in a state machine: as mentioned above, a Behavior of a Transition is executed if the Transition is fired. It is unclear in the Superstructure Specification what happens, if there is already an existing behavior invocation of the Behavior on a Transition and this Behavior is not reentrant. In our formal semantics the Transition will not fire in this case.

**SIMULATION OF UML 2 DIAGRAMS - ActiveChartsIDE**

By the seamlessly reuse of the diagrams for the implementation and the possibility to use regular code within the UML 2 diagrams the creation of applications is simplified. This leads to a continuous development process from the earlier phases to implementation. The creation of software is benefited by simulating and visualizing diagrams in early software development processes for quality assurance.

We extend the ActiveChartsIDE from activity diagrams to other behavior specifications of the UML 2 like state machines and sequence charts. The unnecessary restriction to activity diagrams is canceled this way. The use of different kinds of specification as described in the Introduction (see figure 1) clearly benefits the development of applications.

**Figure 4:** Possible definition of test cases

Sequence diagrams can be used to define test cases, for example. Figure 4 shows a simplified test case for the MP3-player: a possible interaction of a user and the player is illustrated and the reaction of the player is defined by the behavior executions on its lifetime.

The usage of hand-written Code within the models was extended for the afore mentioned diagram types. Furthermore, the advantage of having mathematically precise formulated semantics is that rigorous analysis can be performed on the models. To ensure quality we enriched the tool with syntactic checks on the models. Having a formal semantics allows us to perform semantic checks on the models, even.

**CONCLUSION AND RELATED WORK**

In this paper we introduced a formal semantics for combining different behavioral specifications in the UML 2 using Abstract State Machines. We need a precise semantics for UML 2 diagrams to allow interpretation by the ActiveChartsIDE. Based on formal semantics for the individual language units we specified rules to call behaviors from other behavioral specifications.

ASMs are appropriate to formalize the semantics of modeling languages. They are based on a solid theoretical foundation (evolving algebras) and the syntax is understandable for users with programming background [7] [18]. In addition, there is tool support for analysis and running ASMs and an ASM formalization provides a good starting point for implementations.

Different groups work on the formalization for the UML. [27] [26] [25] and [21] formalized the semantics for activities. These approaches work on the formalization of individual language units. [21] includes a formalized semantics for integrating code into models but supports only activities as valid behaviors in a CallBehaviorAction. The UML semantics project [28] was established to create a mathematically formalized semantics for UML. Based on the so called System-Model actions and activities are integrated in a semantic model for the UML 2.0 [5] [6]. The three-layer architecture of the UML is implemented this way. [13] provides a formal semantics for UML 1.4 subsystems which allows them to interact by message passing.

In our approach we explicitly look at the combination of behavioral specifications. We use the different existing formalizations for the language units and specify a semantics for interacting UML behavior specifications. We concentrate on the top level of the semantics layers and do not establish a complete semantics for the lower levels like Actions.

In the future we will integrate other formalized specifications like Use Cases into the ActiveChartsIDE. There is currently an ongoing master's thesis investigating consistency aspects of UML models. Moreover, we analyze the effect of time at transitions in state machines.
References


**BIOGRAPHY**

**Jens Kuhlmeier** was born in Augsburg, Germany and went to Ulm University. He received his Diploma in Computer Science in 2003. He works at the Institute of Software Engineering and Compiler Construction for five years now. His research interests are formal methods, modeling, simulation and the UML.
SIMULATION TOOLS
SOFTWARE TOOLS
DISPATCHING INDEPENDENT RAILWAY SIMULATIONS WITH DIRS:
A SIMPLE RSH-BASED TOOL

Damien Feillé
LIMOS UMR CNRS 6158
SNCF Innovation and Research
45 rue de Londres 75379 Paris Cedex 8
e-mail : damien.feillee@gmail.com

David Hill
LIMOS UMR CNRS 6158
Blaise Pascal University
BP 10125 - 63177 AUBIERE Cedex
e-mail : drch@isima.fr

Gilles Dessagne
SNCF Innovation and Research
45 rue de Londres 75379 Paris Cedex 8
e-mail : gilles.dessagne@sncf.fr

KEYS WORDS
Railway simulation, Distributed Simulations, Experimental plans.

ABSTRACT
The evaluation of the impact of incidents on the robustness of a timetable is a very important issue at the French National Railway Society (SNCF). A railway simulator has been coupled to a statistical tool in order to obtain consistent indicators. An experimental plan, proposing a number of combinations of the input parameters often implies a large computing. An automatic solution enabling the distribution of simulations on an heterogeneous set of computers has been developed to analyze different railway system configurations.

INTRODUCTION
In order to increase the understanding of train behaviors on their railway system, the SNCF has developed a simulator named SISYFE (Fontaine and Gauyaq 2001). It simulates precisely various configurations of a railway system composed of an infrastructure, a system of signalization, the traffic: a list of human operators with the material they handle (trains), a schedule (timetable), a regulation method (scheduling of the trains and route selection) and a scenario of incidents. SISYFE is a deterministic discrete events simulator. The time management is clock based or event based. SISYFE calculates finely train dynamics; it models the signalization, the behavior of the train operator and manages the exploitation of incidents of various types (delay, breakdowns related to the infrastructure, etc.). This tool is mainly used:

- to evaluate the robustness of schedules,
- to evaluate feasibility studies,
- to help in defining exploitation strategies,
- to validate supervision tools,
- to calculate operating parameters for optimization tools,
- to compute the electric power consumption, etc.

Since the beginning of computer simulation, experimental design has been an active research field for improving the effectiveness of simulated systems (Kempthorne 1952) (Zeigler 1976; Kleijn 1987; Balci and Sargent 1989). It still remains a valuable technique in the decision-making context (Kleijn and Groenendaal 1992; Hill 1996; Amblard et al 2003). In order to obtain pertinent results, a rigorous methodology of experimental designs was considered and we used the SARDAIGNE tool (Chandesris and Labouisse 2003).

This tool enables to design a sound experimental plan starting from a set of railway system configurations and real data (giving the nature and the frequency of the incidents). The experimental plan is designed to provide an “optimal” set of independent simulations. Each independent simulation runs in less than one hour on current personal computers. Experimental plans often lead to several thousand of independent simulations which have to be run. At the end of the computing of these simulations, indicators are generated to analyze the traffic plan robustness (considering a reasonable amount of small incidents). The fact that the SARDAIGNE tool takes into account the statistical occurrence of small incidents, and generates an experimental plan with deterministic events, enables to forget stochastic aspects in the SISYFE simulator, which thus remains entirely deterministic. However, scientists at SNCF have to deal with the computing burden linked to the execution of thousands of simulations. Such simulations can be considered as a pertinent set of replications, which carefully studies the scenarios considered important by SNCF experts.

SIMULATION OF THE TIMETABLE AND STATISTICAL ANALYSIS
As seen in the introduction, SISYFE (Fontaine and Gauyaq 2001) can work on a set of quite distinct data. For the simulation of the experimental designs, a tree of assumptions is generated in order to store the various configurations (Figure 1: Assumptions tree).

The SARDAIGNE tool (Chandesris and Labouisse 2003) is used before and after the simulator. Indeed, the first use is
dedicated to the generation of the tree of assumptions and to the generation of events related to traffic incidents. A second use of SARDAIGNE is done to post-process simulation results and to provide statistical indicators (Figure 2: Connection between SISYFE and SARDAIGNE).

![Assumptions tree diagram]

**Figure 1: Assumptions tree**

**PROBLEM OF COMPUTING TIME**

The various information gathered are combined in an assumptions tree gathering the whole set of data needed to the experimentation. By taking into account all the possible combinations, it is frequent that an experimental design leads to several thousands of simulations.

The duration of a simulation depends mainly on the complexity of the data on which it works (for example, a very wide network or a very long mission will increase considerably the computing time). For instance, small networks simulated for a few hours of virtual time can be computed in less than half an hour, whereas larger networks simulating 24 hours of virtual time can necessitate a few hours of computing on current personal computers. If we have only one computer at our disposal a few weeks of computing is needed. With this kind of independent computing, it is obvious to consider a parallel implementation. A cluster of PCs can be considered or even an access to a computing grid (Foster and Kesselman 1999).

In the small research unit where the programs are developed, such parallel architectures could not be retained. Thus, we have to consider virtualization techniques to access a parallel virtual machine in a similar way to what was proposed in PVM (Geist et al 1994). We have designed a distribution framework dedicated to a small network of heterogeneous machines which could run either Unix/Linux or Windows. The only constraints were that a target machine had to run a Remote Shell daemon (rsh). We could have considered batch systems and jobs scheduling programs such as PBS (Portable Batch System), LSF or other batch frameworks, but we were not able to handle the heterogeneity of operating system versions found in our research unit. In addition, we had only one application to dispatch and we have preferred carefully crafted distribution software to obtain the maximum speedup.

![Diagram of SISYFE and SARDAIGNE connection]

**Figure 2: Connection between SISYFE and SARDAIGNE**

**SEVERAL MODE OF DISTRIBUTION**

Within the framework of our railway simulator, three distribution techniques are possible:

1. A distribution of independent experiments.
2. A spatial distribution.
3. A functional distribution.
If we retain the first technique, executions of the simulator are carried out with different input parameters. This technique is very simple to set up when simulations are completely independent from each other. Running the system under different configurations helps in the validation phase and improves the quality of statistical results. In addition, the distribution of independent simulations always gives a very interesting speedup. We have mainly retained this technique for our application.

In the case of a spatial distribution, several simulators run in parallel and spatially cooperate for a single simulation.

Each simulator is responsible for a part of the network and manages only the trains which are in its zone. This technique will be used in future developments. For a functional distribution, the simulation is limited to a specific number of functions. It becomes interesting when the communications time is smaller than the computing time required to simulate elementary functions. Such functions, taking into account particular aspects of a train’s march often requires significant computing. This solution will also be used in the future.

![Diagram](image)

**Figure 3**: Elementary functioning of the scheduler

**CONCRETE DISTRIBUTION OF SIMULATIONS**

The problem we have presented is naturally parallel, each simulation of the experimental design being independent from the others. We have chosen and automatic distribution of the computing load and this is carried out by a scheduler. The scheduler is placed on a master calculator and dispatches the whole of simulations on a set of slave computers. The scheduler manages only one queue and has to ensure the correct operation of each execution. It sends a new simulation to any available slaves when it is possible. Simulations which have failed can be re-started at the end of a first round of dispatching in order to finish the experimental campaign.
Like in PVM, a virtual machine is built with a simple list of machine names. The parameterization of this list is easily achieved using a file skeleton with keywords. This configuration file is necessary for the good course of the whole simulation. For each computer embedded in the virtual machine, the set of information contained in this file are the following: 
- the operating system of the computer,
- the launching directory,
- the account information (login and passwd),
- a fixed number of authorized executions,
- a schedule to specify when it is authorized to launch a simulation.

With the configuration file several launching slots are created, a slot is a space window enabling the launching of a simulation by the scheduler. The scheduler operation simply following this rule: as long as there remains a free slot, a new simulation is started in this slot. When all the slots are full, the system waits until there is a free slot to pass on the next simulation. This is done until all simulations have been executed.

**DIFFERENT EXECUTION PHASES**

The execution of a single simulation possesses several phases. We have to setup the tree structure and the data necessary to launch a simulation (selection of the right simulation data, parameters settings, …). This tree structure is then sent on the machine where the simulation will be computed. We use a simple rsh mechanism to set data on the computer, data are first compressed and next sent with the following command: “tar cf -<directory> | rsh -l <user> <computer> tar xf - “. A right configuration of rsh access on each computer is necessary in order to have an access without password (on UNIX with the .rhosts file). The package which is sent on the machine contains:
- data of the simulation (network, trains, timetable, …)
- executable of the simulator
- launching script of the simulator (set of various environment variable, …)
- configuration file of the simulator

We start the simulation on the remote host with a rsh command that runs a script shell previously uploaded on the computer in order to set all the global variables and to launch the simulator.

We collect the results which are transferred on the master computer with a FTP batch script command. Results are saved by default on the master computer or on the selected computer. The disk space used by a simulation is cleaned with an rsh command which removes all launch data on the computing computer. When using the mode which enables the automatic storage of all the result files, we have setup a MySQL database.

**GLOBAL TOOL DESCRIPTION**

Several versions of the tools have been developed. In a first time we developed a script method (in Perl and Shell scripts) to validate the concept. Next, a Java interface has been developed to give more flexibility to users. The finished interface permits to configure slave computers access, to precise the authorized launching time, to select the simulations to distributed with many parameters (simulation results traces, various output files,…). Failed simulations can be automatically restarted at the end of the execution. In order to identify bad simulations (abnormal runs), a watch dog system has been added. Thus we can kill simulations which take too much time, to identify simulations which do not last enough time and to identify “bad” slave computers (if 3 simulations failed successively on the same computer, this slave is disabled and no simulations will be launched on it).

A console view of all executions helps in managing the simulations. We can display which simulation runs on which slave computer, the current computing time for each execution and the foreseeable end of the execution (computing on pass simulation). We can stop manually each execution and we can also stop all simulations.

Finally, we can also predict the approximate time when all simulations will be finished (if we do not have too much troubles). At the end of all simulations an e-mail with the information of the execution is sent to the user. The e-mail contains:
- the total computing time of the study,
- the number of failed simulations,
- the number of successful simulations,
- the computing time needed for each simulation,
- the launching parameters (computer, start time, …) of each simulation.

**DISCUSSION**

The tool developed enables an automatic dispatching of simulations and we obtained a very interesting speedup in a small and heterogeneous computing environment. The scheduling of simulation jobs is of course very efficient at night with the maximum expected speedup, i.e. we divide the computing time by the number of slot. Indeed, the transfer time is negligible compared to the pure computing time. For instance, before the implementation of this tool, the computing of a benchmark using the Baudrecourt network (French East Fast Line) would have taken more than 1000 hours of computing (around 20 minutes for a single simulation, but 3200 independent simulations in the experimental framework). With the dispatching tool and only 12 available slots, the computation of the whole set of simulations took less than 100 hours, very close to the maximum speedup (Figure 4). (the difference is due to the transfer time)
Figure 4: Speedup of the tool

FEEDBACK

The tool permits the computing of bigger experimental plans with much more simulations. The statistical indicators are more precise and we were able to obtain better optimization results... This basic dispatching technique is easy to adapt to other software, in particular we will compute different regulation solutions with the LIPARI software (Lerin and Joie 2004). The dispatching tool includes now several statistics after the simulation. It permit the dispatching of work which needs a lot of computing time like the creation of graphical results with simulation traces stored in the result database.

CONCLUSION

We have proposed a simple rsh based tool to dispatch simulations on a small network of heterogeneous computers and work stations. This tool is easily adaptable and its basic mechanisms can be implemented on every system running Java. The only adaptation will be the script file which has to be adapted to the target operating system. This tool can be seen as a desktop grid solution, but the development of a new set of tools for railway simulation is envisaged for the next years. They will natively embed this kind of simple dispatching technique. The concepts presented here should be improved and automated to reduce human intervention. The final goal is that this kind of distribution should be transparent to final users. Two different kinds of distribution will be possible in the new kind of train simulator we are preparing. A spatial distribution and a distribution of the computing of train’s march are considered. Indeed, in the next few years the French railway system will be separated in different regulation zone, each zone will be independent and will simulate the regional railway configuration. Our new kind of simulator will allow a spatial distribution and functional distribution to compute the train’s march. The new traffic simulator will receive data from the real railway system and will work in real time to compute the estimated transport’s plan.

REFERENCES


AUTHOR BIOGRAPHY

DAMIEN FEILLEE is a PhD student at the LIMOS (ISIMA) of Blaise Pascal University of Clermont Ferrand (France). His work, in cooperation with the SNCF, is focused on discrete events simulation and multi agent system in railway simulation.

e-mail : damien.feillee@gmail.com

DAVID HILL is currently Vice President of Blaise Pascal University in charge of Computer Science and past co-director of ISIMA Computer Science & Modeling Institute. Since 1990, Professor Hill has authored or co-authored more than a hundred technical papers and journal papers and he has published text books.

e-mail : drch@isima.fr

GILLES DESSAGNE is a project manager at the Innovation and Research Department of the SNCF. He holds a PhD in Computer Science from the University Paris 6 (2000). His research fields concern combinatorial linear optimization and applications in transportation networks. He is currently working on a research project focusing on simulation and optimization of railway traffic.

e-mail : gilles.dessagne@sncf.fr
SIMULATION FOR AN OPERATOR’S DECISIONS IN SCADA

Andrzej UR3ANIAK
Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60-965 POZNAŃ,
POLAND
E-mail: andrzej.urbianiak@put.poznan.pl

KEYWORDS
AI in simulation, Decision support systems, Model evaluation, AI-supported simulation, Interactive model

ABSTRACT
In the paper there are analyzed decision support systems for operators of complex processes. Main topic of the paper is artificial intelligence (AI) methods’ utilization in the different levels of decision process. There are described new hybrid architectures as a proposal for modelling of complex processes and for effective support of operators. There are many simulation experiments used for analysis and choice of the proper AI methods for particular steps of these architectures. The elaborated system is dedicated especially for wastewater treatment processes. For evaluation of the developed system there are the well-known ASM2 models used.

INTRODUCTION
Uncertain environments or ill-defined systems are very difficult to model and control, therefore, conventional approaches are usually not adequate for such tasks. In contrast, solutions based on artificial intelligence (AI) approaches are much more successful in managing uncertainty and taking decisions even if not all information is available. Neural network, fuzzy logic and expert systems play an important role in dealing with uncertainty. Moreover, evolutionary algorithms are a robust optimisation technique that can find an acceptable solution for complex problems quickly and reliably. Although artificial intelligence approaches have advantages, they also have their limitations. The solution is applying AI approaches in managing complex processes in combination of two or more AI approaches together in such a way that it would preserve the advantages and minimise the disadvantages.

The proposed simulation system utilises a new architecture, that allows to build a hybrid system using available soft computing methodologies (such as artificial neural network, fuzzy logic and evolutionary algorithms) (Oborzyński2003).

SYSTEM’S ARCHITECTURE
The architecture of proposed system named IFENKAS (Integrated Fuzzy Evolutionary Neural Knowledge/Model based Adaptive System) consists of five subsystems, namely:

- **Initialisation subsystem** – creates an initial model according to the setup information given by the model designer;
- **Model Development subsystem** – develops the model according to the requirements and constraints imposed by the model designer;
- **Model Assessment subsystem** – evaluates the suitability of the model provided as an input;
- **Model Optimisation subsystem** – optimises the model to best meet the requirements and constraints imposed by the model designer;
- **Model Complexity Reduction subsystem** – simplifies the model by searching for elements that do not contribute to the final output.

The basic elements of the IFENKAS architecture and data flow between these elements are presented in Fig.1. This architecture is a general framework that only specifies main subsystems involved in the process of model identification and their role in this process. It does not specify how those systems should be implemented, i.e. how each subsystem should be followed to achieve a desired goal. This feature gives a high degree of flexibility in choosing solutions suitable for a given problem domain.

The most objective implementation of the IFENKAS architecture would be building a system that is able to produce the most effective model from the available data. Moreover, it should be possible to apply such a system in a vast rage of modelling and control problems with an effective degree of computational complexity. The addressed problems are of a complex nature and they are ill-defined. In general, the requirements of the proposed system are twofold. On one hand, it should be able to model high-dimensional systems using readable model with good accuracy. On the other hand, the resulting model should be easy to understand.

The proposed architecture utilizes chosen artificial intelligent approaches respect to functionality of particular subsystems. In the proposed implementation are used the following approaches:

- **Initialisation subsystem** – random knowledge base generation;
- **Model Development subsystem** – co-evolutionary algorithm;
- **Model Assessment subsystem** – fuzzy logic;
- **Model Optimisation subsystem** – neural network;
- **Model Complexity Reduction subsystem** – genetic algorithm.
The selection of respective AI approaches was made on the basis of large data analysis. The most important criteria were the number of data and its accuracy. It is natural that the method’s choice strongly depends on the modelled and controlled process, too.

**IMPLEMENTATION**

An elaborated architecture can be used for several important problems’ solution, especially for modelling of complex systems and its control. There were made the assumptions in this implementation, important for the problems mentioned above. First was the high level of problem complexity and second was uncertainty of system. In the paper, main focus was placed in the artificial methods’ utilisation, thus we decided to use the models well-known in the environmental engineering. Respectively, the proposed methodology were developed for analytical model of a wastewater treatment plant elaborated for International Association on Water Quality (IAWQ) by experts group directed by M. Henze, named ASM 2 (Henzel et al. 1994, 1995). The model is based on the mass balances of the components found in influent water. If the tank of volume $V$ with one inflow and one out flow is considered, then an overall mass balance can be written as:

$$\frac{d(VC)}{dt} = Q_{in} C_{in} - Q_{out} C_{out} \quad (1)$$

where $V$ is a tank volume, $C$ is concentration of the process component in a tank, $Q_{in}$ is influent flow rate, $Q_{out}$ is effluent flow rate, $C_{in}$ is concentration of the component $C$ in influent water and $C_{out}$ is concentration of the component $C$ in effluent water.

However, modelling of wastewater treatment plant processes is more complex because they involve chemical, biological and biochemical reactions. As a result, the general equation describing the mass balance of the component in the wastewater includes also the reaction rate $R_C$ for the specific component in a tank, as shown in equation (2):

$$\frac{d(C)}{dt} = \frac{Q_{in}}{V} (C_{in} - C) + R_C \quad (2)$$

For each specific components we can obtain the equation (like (2)) but with special state variables and parameters concerning analysed component. For example, one of the main undesirable components of effluent water in wastewater treatment plant is the soluble ammonium - $S_{NH_3}$. Thus the equation (2) will describe the derivate of $S_{NH_3}$ (Oborzyński 2003).

The model of the plant includes bioreactor and secondary clarifier. The bioreactor is divided into 10 completely mixed reactors: 5 aerobic and 5 anoxic ones. The secondary clarifier consists of ten vertical layers. For each layer the respective one-dimensional differential
equations are formulated (Watson et al. 1994, Jeppsson 1996).

The model was implemented in C# using Microsoft Visual Studio.NET environment.

**EXPERIMENT’S METHODOLOGY**

The developed model was tested by taking into account three different functions: modelling, control and classification. Respect to this contribution the most important part of the tests were connected with modelling and control. The modelling of wastewater treatment processes were analysed for soluble ammonium concentration.

The experiment consists of two test cases. In the first test case, input data are generated when influent flow of water is perturbed to mimic the unexpected conditions. In the second test case, influent water characteristics are taken from the real wastewater treatment plant data given by benchmark (COST624, 2002). Started parameters were assumed with respect to remarks given by Henze’s expert group.

For the effective evaluation of proposed hybrid solution there were used the training and testing data sets. The integrated model of the analysed wastewater treatment plant includes 9 input variables using simple knowledge base with three rules. The developed approach was compared with well-known ANFIS model (Jang 1993). In the ANFIS, with the same 9 input variables the number of rules in data base increases drastically.

In the test case, the training set consisted of 500 data records, whereas the testing set consisted of 700 records. In general, the IFENKAS implementation was able to find very small accurate models but in longer time than other approaches (Oborzyński 2003).

There are given the large numerical input data together with obtained results (Oborzyński 2003).

**FINAL REMARKS**

1. The developed IFENKAS system utilises the artificial intelligence methods in proposed architecture respect to their properties and restrictions.

2. The functional extended architecture allows to use different artificial intelligent methods in particular architecture elements.

3. The proposed approach can be utilised for modelling of a wastewater treatment plant and can check in the effects of operator’s decisions by simulation of processes. Thus, the proposed operator’s decision may be simulated at first and after the evaluation may be implemented in a real process or not.

4. The other applications of IFENKAS system concern the control of a wastewater treatment plant. This problem was out of scope in this paper but research is being carried out in this field confirming the usefulness of developed system for the mentioned tasks.

5. On the basis of the proposed approach there is future research planned, especially, towards its extended functionality for managing complex processes.

**REFERENCES**


Oborzyński K. 2003. Integration of artificial intelligence methods in management and control systems of complex processes, PhD Dissertation, Poznań University of Technology, Poznań (Poland)


**BIOGRAPHY**

Andrzej URBANIAK was born in Poland and studied at Poznań University of Technology (control engineering) and Adam Mickiewicz University of Poznań (mathematics) and finished these studies in 1972. He obtained the PhD degree in 1979 and finished his postdoctoral thesis in 1990 at Warsaw university of Technology. From 1990 he has been taking over the position of professor of Institute of Computing Science. He is an author and co-author of 5 books and over 250 papers concerning computer control systems and application of information technology in environmental engineering. He is a reviewer of several scientific papers and member of scientific committees of international conferences.
FROM DESIGN WITH SYSML TO VHDL-AMS SIMULATION

Jean Verries, Mario Paludetto, Abd-El-Kader Sahraoui,
LAAS-CNRS and University of Toulouse
7 Avenue Colonel Roche,
31077 Toulouse
Email: jverries@laas.fr

ABSTRACT

An approach combining SysML and VHDL-AMS is proposed in this paper. The design is modeled with SysML and then we derive some intuitive rules to obtain the VHDL-AMS model of the lower level blocks built in SysML. The work is at the level of the tentative approach that is being carried out on real industrial application for onboard systems. The paper goes beyond the models issues and carries out the simulation procedure that are available on tools to validate the design for the intended blocks.

INTRODUCTION

Model Based System Engineering tends to provides designer with languages and tools to depict analysis, requirement and design artefacts, and to relate them by traceability links. Expected benefit is to improve communication between stakeholders with removing ambiguity and improve completeness, better management of system complexity and earlier integration of V&V activities. On the other side, powerful modelling language enables simulation at system level, allowing designers to improve design confidence and maturity within and across projects at early stage of development. However, modelling in itself can be a complex and costly task, thus reducing expected benefits.

The work is carried out in the context of deploying systems engineering practice for aeronautics equipment subsystems. The processes have been defined from the requirements management process, the design process, implementation and validation/verification through simulation. The project is under constraints on aeronautics standards. The paper focuses on methods and tools.

In this paper, we present how SysML could be used to build VHDL-AMS model and thus provide an efficient way to model and simulate systems at architectural level. We start by short presentation of SysML’s and VHDL-AMS’s subsets that we focus on, and the rationales for their integration into a single framework. We propose several steps that designer has to follow, in order to benefit both from SysML and VHDL-AMS contribution for system design modelling techniques. This integration covers a broad range in the system development, starting from requirement expression, architecture alternative proposal, and technical performance assessment. We show how bidirectional mapping can be established between architectural descriptions in SysML and structure of VHDL-AMS models, and thus automating partly the modelling process.

SYSML AND VHDL-AMS PRESENTATION

SysML general presentation. SysML (System Modelling Language) is a general purpose, graphical modelling language for system engineering. It allows analysis, specification and design of systems. Using SysML, system designer or design team can graphically depicts system operational context and use cases, maintain structured set of requirements, model behaviour, system logical and physical structure, and realise all association link between these artefacts to ensure a seamless flow from initial analysis to detailed design. In particular, SysML can add great benefit to validation and verification planning and support, allowing designers to directly trace these activities against system models and system requirements. Benefits on design process are an improved communication with a model-centric approach, improve validation and verification activities by relating them to requirement engineering and logical and physical design.

Initially, SysML results of a decision of INCOSE in 2001 to cast UML for system engineering specific domain. Then, INCOSE and OMG have jointly decided to create a working group, to specify requirements for a system modelling language. This led to define UML for SE RFP [1], a requirement set that specify need for a system modelling language.

UML for SE RFP has lead to the definition of a SysML draft in 2004, by SysML partner, an association of majors industry actors and tool vendors. Then, a first version of SysML has been submitted to OMG and adopted in 2005. Some competing versions was proposed, and a merging of them was finally adopted by OMG in 2006. OMG SysML v1.0 [2] is available as an open source available specification since September 2007.

SysML constructs and diagrams. This section presents briefly SysML constructs used in this paper. First, SysML blocks and block diagrams, that allows designer to easily depict architecture, from basics concepts (for example few interconnected block that represents one efficient solution in a specific context) to a detailed, component-level implementation.

Blocks are the modular elements of system descriptions. Each block defines a collection of features to describe a system or part of it. These may include both structural and behavioural features, such as properties and operations, to represent the state and modes that the system may exhibit. Blocks provide a general-purpose capability to model systems as trees of modular components. The specific kinds of components, the kinds of connections between them, and the way these elements combine to define the total system
can all be selected according to the goals of a particular system model. SysML blocks can be used throughout all phases of system specification and design, and can be applied to many different kinds of systems. These include modelling either the logical or physical decomposition of a system, and the specification of software, hardware, or human elements. Parts in these systems may interact by many different means, such as software operations, discrete state transitions, flows of inputs and outputs, or continuous interactions. Block can have multiple compartments allowing to describe its features. For example, structure compartment show elements that appear in an internal block diagram, as described below.

Block can be interconnected in many ways and appear in two major diagram types: Definition Diagram, and Internal Block Diagram of SysML, depicts respectively component structural hierarchy and interconnections.

Block Definition diagram is based on UML class diagram, with several restrictions and extensions. The Block Definition Diagram in SysML defines features of blocks and relationships between blocks such as associations, generalizations, and dependencies. It captures the definition of blocks in terms of properties and operations, and relationships such as a system hierarchy or a system classification tree. The Internal Block Diagram captures the internal structure of a block in terms of properties and connectors between properties. It depicts flows between system components, that can be logical or physical: service, data, energy, matter, or combination of them.

Other main construct provided by SysML is Requirement block. Requirements are modelled as an extension of UML class. Requirement blocks allow to specify textual requirements, and identify it with a unique identifier. Others attributes may be associated to state validation/verification attributes and method or other information on requirement life-cycle. Main interest of SysML requirements is that several relations can be established with others requirement and SysML modelling artefacts. Requirements can be related to others requirements by refinement relation, thus enabling to depict requirement flow down for each design level. Also, they can be connected to design block by “satisfaction” link, stating that design block satisfy requirement.

While SysML can bring benefits to system designer, it has voids that can be identified by comparing its specification to the original RFP. Following are some limitations and voids that are addressed in this paper: Although parametric diagrams allows expression of mathematical relations, SysML does not provide support for modelling of mixed-signal system description. Moreover, although simulation scenarios could be represented as test case that verify requirements, no mean is explicitly provided to present simulation context, objectives, scenarios and results. Lastly, no explicit artefact or language stereotype has been added to model design alternatives, neither mean to assist assessment and selection of design alternative.

The VHDL-AMS language

VHDL-AMS is a hardware description language. Based on VHDL language, it has been developed to extend VHDL to the description and the simulation of analog, digital, and mixed-signal systems. VHDL-AMS has been normalised (IEEE standard 1076-1993) as an extension of VHDL language. Thus, VHDL can be considered as a subset of VHDL-AMS language. The first release of the IEEE 1076.1 standard has been available since 1999. Following section present briefly the main advantages and characteristic of VHDL-AMS.

One of the major benefits of VHDL-AMS is its ability to easily model and simulate systems, that include different physical domains such as electric, mechanic. System behaviour can be modelled via acausal equations, and therefore facilitate reuse of components in different contexts of use. Also, VHDL-AMS allows designer to model system at different abstraction level, thus improving performance and simulation, and improving overall cost/benefit ratio of modelling and simulation activities. Due to this multi-abstraction capability, VHDL-AMS can be used throughout all system development life cycle, from architectural exploration and conceptualisation, down to single component functional finest, structural modelling.

Continuous and event driven modelling brings many advantages in system modelling: Many physical systems have different sets of equations depending on their operating domain. Using event concepts, development of these models can be simplified. Conservative physical system, event driven behaviours, logics and analogic signal conditioning can also be gathered on same system model. Then, VHDL-AMS can be a powerful tool for system designer.

Other characteristic of VHDL-AMS, inherited from VHDL, is its capability to define multiple implementations of same component interface. Indirect component instantiation and configuration brings genericity support, enabling designer to implement many design alternatives for same interface, different abstractions levels or modelling concerns in complex model structures. In early phase of top down design process, this capability assist designer in the product logical and physical decomposition. This decomposition is done iteratively until definition of elementary components, which are described in their functional, behavioural or physical aspect. Each component is defined by only one model, and instantiated as many as necessary in the global model.

MODEL SPECIFICATION METHOD

This section describes general model-based system design process, around architectural alternative assessment.

When developing hybrid systems in terms of mixed mechanical, electronic equipment, there is a need for a high level of abstraction when devising architecture. The system view is preferred for its effectiveness in tackling such types of systems. The choice of SysML is meant to have an independent method rather than choosing specific
technology method; also VHDL-AMS is a general purpose notation for hybrid systems even it is known that it was devised initially for electronics systems.

During physical solution definition, some functional and performance requirements are identified as key architecture efficiency indicators. These indicators will be assessed by an executable model, by translating them into a set of measurable values on physical model. Two goals must be satisfied when building such a model:

First, try to completely simulate parameters that have been identified as key efficiency indicators. This corresponds to a top-down view of the model specification, starting from high level, stakeholder needs, and allows to ensure that purpose of system is done according to functional specification.

Next, allow identification of unexpected or undesirable effect that can lead to reject an architecture. This issue is tightly linked to system internal and external interfaces, and operating environment. As it is depends on technical solutions, this corresponds to the bottom-up aspect of model specification. (For example, one model does not simulate heat transfer between two part of system, resulting in unrealistic gas temperature in pneumatic actuator, and therefore unrealistic operating performance).

Following steps propose a way to specify and build physical models trying to improve efficiency and benefits of modelling and simulation tasks.

1. Identify technical effectiveness metrics on Logical Architecture Solutions. Architecture effectiveness metrics should be expressed in a solution independent point-of-view. This effectiveness metrics should be approved by stakeholders, for example during logical solution review. In SysML, we specify attributes to component block in order to specify internal values that have to be simulated in the dynamic executable model. For example, electrical consumption, speed profile, mechanical effort). Expected discrete event properties are specified as sequence or activity diagrams that will be compared to simulation results (for example: aural warning triggering, sensor measurement time).

2. Allocate effectiveness metrics on system components and interfaces. As alternative architectures are explored, efficiency metrics have to be translated and allocated on system parts. Such characteristics are key performance parameters such as effort/torque, speed, response time, hydraulic pressure... These are considered key characteristic in that they are directly traceable against technical efficiency criteria and stakeholders expectations. This allocation process can be based on engineering judgement, or based on trade-off analysis. Exploring design alternative will usually bring to refine or complete set of efficiency metrics previously defined. This is not an issue as long as set of design alternatives refers to the same efficiency metrics reference. For example, assessing one electro-mechanical system against a human powered system can bring designers to assess system energetic autonomy.

3. Specifying simulation sequence and stimulus. In conjunction with effectiveness metrics allocation, one should define simulation conditions, stimuli, and measuring means to ensure that simulation will provide expected benefits. This step is tightly coupled with architecture definitions and may require to develop some additional model parts. For example measuring a numeric response time on a continuous signal shall require developing a measuring component with measured signal being compared to thresholds values and returning required response time value.

4. Derive components internal parameters from key physical characteristics. This task has a great impact on model accuracy. Once key characteristics have been allocated, one should consider component internal parameters that could impact its key characteristics. This is actually a bottom-up analysis, in that it highly depends on intrinsic, physical structure of each component. It is usually performed by engineering judgment, and requires a careful analysis of both the component intrinsic properties and its operating conditions and environment. For example, consider one component as a mechanical damper used in an emergency mechanical system. In this example efficiency metrics naturally brings to allocate a minimum damping effort to this component. Then, the use of a hydraulic actuator should bring to add the oil temperature as an internal parameter to be monitored as it has a great impact on damping effort which will be produced.

5. Identify additional parameters to raise undesired effect simulation. Such task should be derived by engineering analysis such as safety and maintainability analysis. It should also result from a bottom-up analysis of previously identified key physical characteristics and internal parameter.

6. Model dynamic behaviour. In this task, modeller should ensure that instructions that models dynamic behaviour covers computation of key, internal parameters, and also particular parameters. Depending on model abstraction level, these instructions can be differential or algebraic, conservative or non-conservative equations, or transfer functions.

FROM SYSML SYSTEM STRUCTURE TO VHDL-AMS MODEL

We present here a method which can be systematically applied to translate a system structural description in SysML into a VHDL AMS code structure. Both are composed of interconnected, “black box” components.

Followings this translation, VHDL AMS components have to be implemented by DAE or others instructions, thus constituting the dynamic part of system architecture modelling.

In this paper, we focus on a general and systematic way of generating vhdl-ams code, as a set of interconnected entity /
architecture design unit, representative of the physical architecture. This implies at least two major steps:

1/ Instantiating VHDL AMS components according to the corresponding instantiated block in SysML.

2/ For each structure level, Connecting instance to each others according to specified port and signal interface in SysML.

Creating Entities and Architectures

We create a stereotype, applicable to block, called “Test Bench”. This Block is the root for the system and environment description. This block will own parts that are the first level of decomposition in the system structure, and iteratively down to the most detailed level. In this decomposition, each part must be typed by a block, which themselves can contain typed parts, thus generating a tree. By this mechanism, all model structure can be generated by a tree traversal method.

Starting from “testbench” block, which is the root leaf of structural hierarchy, we can apply following procedure to generate code structure:

1/ From SysML Block, Generate Corresponding VHDL-AMS
Entity and Architecture:
2/ For all From SysML typed parts within Block:
   2.1/ Generate VHDL-AMS Component Instance
   declaration,
   2.2/ Specify used architecture,
   2.3/ Iterate on typing block of each parts

At architectural level, alternatives can be view as different arrangement of components. In SysML, Two alternatives will be own different parts in its parts compartment, and will share common traceability link toward top level effectiveness metrics or performance parameter. For example, two architecture alternatives will include respectively an inductive sensor, and rotary potentiometer. Both sensors add some weight to mechanical parts, but only the last will introduce a rotational torque to rotational movement due to friction of its internal parts.

Such relations can be modelled as an inheritance link between a functional, implementation-free block which represent the common design characteristics with its effectiveness drivers, and some derived physical alternatives owning theirs own parts. Logical component will typically have some satisfaction links toward set of requirements, providing traceability of requirements onto inherited architecture alternatives block. Some architectural alternatives will inherit from multiple functional blocks, as they are involved in multiple system functions.

Following is a simplified example, showing two design alternatives for performing a position measurement. Two architectures are therefore implemented for the same entity HPMeasure. Selection of current used architecture(HPM_arch2) is realised by the component instantiation in upper level system architecture body

(HPM_arch2) is realised by the component instantiation in upper level system architecture body

Use WORK : ALL;
Entity System is
   -- generic / port
End entity;
Architecture struct_System of System is
   -- internal parameters
Begin
   alt1 : Entity HPMeasure(HPM_alt1);
   -- Other system components instantiations
   End struct_System;
Entity HPMeasure is
   -- generic / port
End entity;
Architecture HPM_alt1 of HPMeasure is
   -- internal parameters
Begin
   Sign_cond : Entity ISSignCond (archL1);
   Induct_ss : Entity InductiveSensor (archL1);
   Target_ss : Entity Target (archL1);
   End HPM_alt1;
Architecture HPM_alt2 of HPMeasure is
   -- internal parameters
Begin
   Rot_Pot : RotaryPotentiometer
   End HPM_alt2;

Fig 1 : VHDL-AMS translation of architecture alternative.

Interconnecting system components

Last section showed how components can be instantiated from SysML structural hierarchy, allowing description of multiple design alternatives. Next step is to realise components interconnection at a given level of structural hierarchy. For this purpose, SysML flowport and signal constructs are translated into port and port map in VHDL AMS instructions.
Let consider a simple example composed of an outer block called ‘System’ owning two parts called ‘HPMeasure’ et ‘SigCond’. First one retrieve a position angle from system environment and transforms it into a voltage value. Second parts retrieve this value and apply a simple processing, providing a binary value as output. This output is then transmitted to system environment. First, outer flowports, which enable communication between system and its environment are simply reported onto VHDL-AMS entity of system. Flowports direction define IN or OUT attribute of associated port declaration.

HPMeasure and SigCond entities are also declared, including its associated flowport input and output. Because two components are interconnected in system architecture, a local signal, called ‘intem’ has to be declared into system architecture. This local signal is typed according to port type of the components (which must be equal). HPMeasure and SigCond are then instantiated in architecture body of system. Then, it is necessary to map formal port (declared into HPMeasure and SigCond entities declaration) onto actual ports that are declared as System entity declaration and local signal declaration. Figure bellow presents SysML and VHDL-AMS translation of same system structure.

Therefore, followings steps has been performed, after entity and architecture declaration, to interconnect components of a system:

1/ From SysML Top Level System Block to lower level components, declare port according to flowport direction and name.
2/ From Flow port and SysML association between components, at each level, declare intermediary signal, typed with SysML flow port type.
3/ Instantiate component into top level architecture, (with optional architecture specification).
4/ Map components formal ports onto actual embedding block and its intermediary signals.

CURRENT APPLICATION

This modelling approach is currently used in context of aeronautic equipment development to transport aircraft passenger doors equipment, for which emergency opening assistance mean is necessary. This type of system presents multi domain system design with strong safety and maintainability requirements. Multiple architecture alternatives include different command, actuators and mechanism coupling are currently assessed against safety, certification and ease-of-use requirements during normal use and inspection.
Fig 3: SysML Structural description of application

**PERSPECTIVES**

We look forward to generalise such approach by different case studies; the methodology can be extended to “systems of systems” as each class in SysML can represent a specific system; the object oriented design paradigm is well suited for systems families (Sahraoui, 2008). Effectively, Research in this area would seek to develop and apply notions from such areas as complex adaptive system and knowledge management and would seek to develop more of a methodological basis for system family architecting and design. Development of a methodological basis for the design and architecting of system families would do much to enhance present abilities to design loosely coupled and virtual organizations and to enable better architectures for these enterprises that would do much to support interoperability and integration; the issue of interoperability in simulation is a critical issue in this context.

**CONCLUSION**

In this paper we have introduced some rules to obtain simulable model from high level, object oriented structural descriptions. Principles described here will be implemented at tool level, along with current and previous work on behavioural model transformation from UML/SysML toward VHDL-AMS. Eventually, this will allow a complete, requirement-driven modelling and simulation methodology for system design.

**REFERENCES**

http://syseng.omg.org/UML_for_SE_RFP.htm

MOG SysML Specification language, v1.0. formal/2007-09-01

PREDICTION TOOLS FOR STUDENT LEARNING ASSESSMENT IN PROFESSIONAL SCHOOLS

Paulo Almeida¹, Paulo Novais² and José Neves²

¹ Centro de Formação Profissional da Indústria de Calçado, São João da Madeira, Portugal
paulo@cfpic.pt

² Departamento de Informática-CCTC The University of Minho, Braga, Portugal
{pjono,jneves}@di.uminho.pt

KEYWORDS
Artificial Intelligence, Multi-Valued Extended Logic Programming, Quality-of-Information, MOODLE.

ABSTRACT

Professional Schools are in need to access technologies and tools that allow the monitoring of a student evolution course, in acquiring a given skill. Furthermore, they need to be able to predict the presentation of the students on a course before they actually sign up, to either provide them with the extra skills required to succeed, or to adapt the course to the students’ level of knowledge.

Based on a knowledge base of student features, the Student Model, a Student Prediction System must be able to produce estimates on whether a student will succeed on a particular course. This tool must rely on a formal methodology for problem solving to estimate a measure of quality-of-information that branches out from students’ profiles, before trying to guess their likelihood of success.

Indeed, this paper presents an approach to design a Student Prediction System, which is, in fact, a reasoner, in the sense that, presented with a new problem description (a student outline) it produces a solved problem, i.e., a diagnostic of the student potential of success.

INTRODUCTION

This work presents an approach to the design of a Student Prediction System (SPS), with a strong emphasis on its reasoning module, endorsed by a process of quantification of the Quality-of-Information (QI) that stems out from the analysis of the student data, here understood in terms of the extensions of the predicates or logical functions that make it. It is intended to produce a Decision Support System (DSS) to help teachers, tutors, psychologists and others, to forecast problems on the evolution of the student state of knowledge, and decide and take the appropriate measures to work them out, on time.

The SPS uses data from several sources to build and evolve the student models. Based on those models it presents the decision makers with a forecast of the student learning success. We start by summarizing the notion of e-Learning Systems and present some features of the MOODLE e-Learning one. Next, we present a methodology for representing data and knowledge in SPS, and evaluate the QI of a student model, as a way to assess the reliability of the information it contains. Finally, we further elaborate on the concept of student models and system beliefs, and our vision of a SPS, including its Student Model Builder and, mainly, its Student Prediction Module.

LEARNING MANAGEMENT SYSTEMS AND MOODLE

Learning Management Systems/Course Management Systems (LMS/CMS) are domain independent, general purpose programmes/platforms, which provide authoring, sequencing, and aggregation tools that structure content, to ease the learning process. It is the duty of the course designer to select and organize the matter in order to build a tutorial module for a given knowledge domain. MOODLE platform (Moodle - A Free, Open Source Course Management System for Online Learning, n.d.) is an instance of a LMS/CMS.

The MOODLE Learning Management System

MOODLE (Modular Object-Oriented Dynamic Learning Environment) has a number of interactive learning activity components like forums, chats, quizzes and assignments. Very interesting is the lesson activity, wherein it is possible to write interactive learning tasks with conditional paths, adapting to the student learning process. In addition, MOODLE includes a logging module to track users’ accesses and the activities and resources that have been accessed. Administrators and teachers can extract reports from this data. Figure 1 presents a high level view of the MOODLE modules.

**Figure 1: The MOODLE LMS Modules**
LMSs should have some sort of knowledge about the students and about their learning processes. This knowledge, i.e., the beliefs the system has about the students’ state of knowledge, is commonly called the Student Model (SM). Indeed, without a SM, the system would simply behave the same way for all students.

**KNOWLEDGE REPRESENTATION FOR INFORMATION QUALITY ASSESSMENT**

Knowledge representation is a crucial factor regarding the success of the operation of a DSS (Neves, 1984; Way, 1991; Analide et al., 2006: 436-442; Ginsberg, 1991).

A suitable representation of incomplete information and uncertainty is needed, one that supports non-monotonic reasoning.

In a classical logical theory, the proof of a theorem results on a true or false truth value, or in an unknown value. On contrary, in a Logic Program (LP), the answer to a question is given by the logical constants true or false. This is a consequence of the limitations of the knowledge representation in a LP, because explicit representation of negative information is not allowed. Additionally, the operational semantics applies the Closed-World Assumption (CWA) (Hustadt, 1994) to all predicates. Usually, LP represents implicitly negative information assuming the application of reasoning according to the CWA.

An extended logic program (Program 1), on the other hand, is a finite collection of rules of the form (Neves, 1984; Gelfond and Lifschitz, 1990):

\[
\text{(1)} \quad \text{q} \leftarrow \text{p}_1 \land \ldots \land \text{p}_m \land \neg \text{p}_{m+1} \land \ldots \land \neg \text{p}_{m+n}
\]

\[
\text{(2)} \quad \text{? p}_1 \land \ldots \land \text{p}_m \land \neg \text{p}_{m+1} \land \ldots \land \neg \text{p}_{m+n}
\]

**Program 1: An Extended Logic Program**

where ? is a domain atom denoting falsity, p and q are classical ground literals, i.e., either positive atoms or atoms preceded by the classical negation sign \(\neg\). Every program is associated with a set of abducibles. Abducibles can be seen as hypotheses that provide possible solutions or explanations for given queries, here given in the form of exceptions to the extensions of the predicates that make the program.

The objective is to provide expressive power for representing explicitly negative information, as well as directly describe the CWA for some predicates, also known as predicate circumscription (Parsons, 1996). Three types of answers to a given question are then possible: true, false and unknown (Neves, 1984). The representation of null values will be scoped by Extended Logic Programming (ELP). We will consider two types of null values: the first will allow for the representation of unknown values, not necessarily from a given set of values; and the second will represent unknown values from a given set of possible values. To see how null values can be used to represent unknown information, let us consider the extensions of some predicates whose attributes resemble that of a SM, namely:

- had_attended: Student \times \text{StrValue}
- motivation: Student \times \text{Value}
- grade_PA: Student \times \text{Value}

where the former argument denotes a given student and the second represents the value of a particular asset (e.g.,

**motivation(ana, 1)** means that the motivation of the student Ana has the logical value 1).

In Program 2, the symbol \(\neg\) denotes strong negation, symbolizing what should be interpreted as false, and the term not designates negation-by-failure.

**Program 2: An extension of the predicate that denotes the motivation of student Ana**

Let us admit that the motivation of another student, say, Diana, has not yet been established. This will be denoted by a null value of the type unknown, as is given in Program 3: the student has some motivation but it is not possible to be certain about its truth value. In the first clause, the symbol \(\bot\) represents a null value of an undefined type. It is a representation that assumes any value as a viable solution, but without being given a clue about which value one is speaking about. It is not possible to compute the value of the motivation of student Diana. The third clause of the program (the closure of predicate motivation) discards the possibility of being assumed as false any question on the specific value of motivation for Diana.

**Program 3: Motivation of student Diana, with an unknown value.**

Let us now consider the case in which the value of the motivation for a certain student is foreseen to be 0.60, with an error margin of 0.15. It is not possible to be positive, concerning the motivation truth value. However, it is false that the student has a motivation value of 0.80 or 1. This example suggests that the lack of knowledge may only be associated to an enumerated set of possible known values. As a different case, let us consider the motivation of the student Paulo, that is unknown, but one knows that it is specifically 0.30 or 0.50 (Program 4).

**Program 4: A logical illustration of the motivations for students Ana, Diana, Carlos and Paulo**

Using ELP, as the logic programming language, a procedure given in terms of the extension of a predicate called demo, is given by Program 5. This predicate allows one to reason about the body of knowledge presented in a particular
domain, set on the formalism referred to above. Given a
question, it returns a solution based on a set of assumptions.
This meta-predicate is defined as:
Demo: Question x Answer
where Question denotes a theorem to be proved and Answer
denotes a truth value: True (T), False (F) or Unknown (U).

demo(Q, T) ← Q
demo(Q, F) ← ¬Q
demo(Q, U) ← not Q ∧ not ¬Q
Program 5: An extension of the meta-predicate demo.

QUALITY-OF-INFORMATION OF THE STUDENT MODELS

Until now, we have seen that SM may not always produce
predictions based only on LP representations of system
beliefs. We have also seen how SM uses ELP to express its
uncertainty and overcome this limitation. In any decision
making process, the decision is made without having all the
information pertaining to the problem. When the decision
maker has to, he/she makes the decision using the available
information, to the best of his/her knowledge. How much a
teacher relies on the predictions at hand? How can SM
provide him/her with a measure of the quality of that
information?

Let i (i ∈ 1 … m) represent the predicates whose extensions
make an extended logic program that models the universe of
discourse, and j (j ∈ 1 … n) the attributes of those
predicates. Let \( x_j \in [\min_j, \max_j] \) be a value for attribute j. To
each predicate is also associated a scoring function, \( V_j: [\min_j, \max_j] \rightarrow 0 \ldots 1 \), that gives the score that predicate i assigns
to a value of attribute j in the range of its acceptable values,
i.e., its domain (for simplicity, scores are kept in the interval
[0 … 1]), here given in the form:

\[
\text{all(attribute_exception_list, sub_expression, invariants)}
\]

This denotes that sub_expression should hold for each
combination of the exceptions of the extensions of the predicates that represent the attributes in the
attribute_exception_list and the invariants.

This is further translated by introducing three new
predicates. The first predicate creates a list of all possible
exception combinations (pairs, triples, ..., n-tuples) as a list
of sets determined by the domain size (and the invariants).
The second predicate recourses through this list and makes a
call to the third predicate for each exception combination.
The third predicate denotes sub_expression, giving, for each
predicate, the respective score function. The Quality-of-
Information (QI) with respect to a generic predicate P is
therefore given by \( QIP = \frac{1}{\text{Card}} \), where Card denotes the
cardinality of the exception set for P, if the exception set is
disjoint. If the exception set is disjoint, the quality of
information is given by:

\[
QIP = \frac{1}{\sum_{i=1}^{\text{Card}} + \cdots + \text{Card}}
\]

where \( C_{\text{Card}} \) is a card-combination subset, with Card
elements.

The next element of the model to be considered is the
relative importance that a predicate assigns to each of its
attributes under observation: \( w_j \) stands for the relevance of
attribute j for predicate i (it is also assumed that the weights
of all predicates are normalized), i.e.:

\[
\forall i \sum_{j=1}^{n} w_j = 1
\]

It is now possible to define a predicate’s scoring function,
i.e., a value \( x = (x_1, \ldots, x_n) \) in the multi-dimensional space
defined by the attributes domains, which is given in the form:

\[
V_i(x) = \sum_{j=1}^{n} w_j \times V_j(x_j)
\]

It is therefore possible to measure the QI that occurs as a
result of a logic program that makes a SM, by posting the
\( V_i(x) \) values into a multi-dimensional space and projecting it
onto a two-dimensional one.

Using this procedure, one gets (Figure 2).

Figure 2: A measure of the QI for a Logic Program or
Theory P

where the dashed n-slices of the circle (built on the
extensions of the predicates \( p_1, \ldots, p_n \)) denote the QI that is
associated with each of the predicate extensions that make
the logic program. It is now possible to evaluate the QI of
the SMs of Ana and Diana (Figures 3 and 4). Let us consider
the logic programs 6 and 7, which represent a set of beliefs
about students Ana and Diana, as well as exceptions to those
beliefs, which are given in the form:

motivation(ana,1)
had_attended(ana, geometry)
¬grade_PA(S, V) ← not grade_PA(S, V),
not exception(grade_PA(S, V))
exception(grade_PA(ana,14))
exception(grade_PA(ana,16))

Program 6: A formal description of the Universe of
Discourse for Ana SM
−motivation(S, V) ← not motivation(S, V),
    not exception(motivation(S, V))
exception(motivation(S, V)) ← motivation(S, ⊥)
−had_attended(S, V) ← not had_attended(S, V),
    not exception(had_attended(S, V))
exception(had_attended(S, V)) ← had_attended(S, ⊥)
−grade_PA(S, V) ← not grade_PA(S, V),
    not exception(grade_PA(S, V))
motivation(diana, ⊥)
had_attended(diana, ⊥)
exception(grade_PA(diana, 8))
exception(grade_PA(diana, 11))

Program 7: A formal description of the Universe of Discourse for Diana SM

In order to find the relationships among the extensions of these predicates, we evaluate the relevance of the QI, given in the form \( V_{motivation}(ana) = 1; V_{grade_PA}(ana) = 0.5; V_{had_attended}(ana) = 1 \). Roughly, this means that we are sure about the motivation and attendance information of Ana; but we are not so sure about the information on the percentage of right answers. As for Diana, we are not sure whatsoever about her motivation and attendance, although we have some assurance (0.5) on her percentage of right answers.

Figure 3: A measure of the Quality-of-Information for Ana

STUDENT MODELS

Therefore, a SM may be defined as a representation of the set of beliefs that a system has about a student (Self, 1994). We will follow Self’s definition but we will use Multi-Valued Extended Logic Programs (MVELP) to express those beliefs, being the truth values bound to a proven theorem given in terms of a composition of the measures of the Quality-of-Information of the predicates that make it (Neves, et al 2007). Indeed, let \( B_A \) denote that a program \( A \) subscribes the substance (i.e., the essence of the extension) of predicate \( p \).

Figure 4: A measure of the Quality-of-Information for Diana

The belief set \( B_A \) configures the extensions of the set of predicates assumed by program A: \( B_A = \{ p \mid B_A p \} \). By applying this line of thought, we can define:

- \( B_S \) as the student set of beliefs;
- \( B_C \) as the computer system set of beliefs. This set includes the extensions of those predicates that the system believes with respect to the general student behaviour; and
- SM as the subset of the system’s beliefs which stand for the beliefs that the system C has about the student S: \( B_C(S) = \{ p \mid B_C p(S) \} \).

The student’s beliefs are not known by the system; therefore, all reasoning about the student has to be made through the analysis of the SM. As for \( B_C \), this set contains the domain knowledge, beliefs about student behaviour, as well as SMs, among others sources of information. The relationship between these sets is shown in Figure 5.

Figure 5: Relationship between \( B_S \), \( B_C \) and SM (Self, 1994)

MOODLE does not have a true SM. It only has a student profile. However, MOODLE does collect metrics about all sorts of actions made by the users. SMs can be built from the history logs of the platform and updated with student activity logs, as we shall see later. In fact, there is a great amount of discussion about the feasibility of SMs from student interactions with learning platforms. Arroyo et al. (2004: 782-784), Johnson et al. (2005) and Mislevy et al. (1999: 437-446), have some work done on this area, mainly through the use of Bayesian Networks in conjunction with Data Mining, to model students’ behaviour.
STUDENT PREDICTION SYSTEM

A SPS configures a DSS made of two modules: a module to create and update the SMs and the system beliefs, $B_C$; and a module to estimate the possibilities of success of the student on a course. The former is the Student Model Builder (SMB); the latter is the Student Prediction Module (SPM). Its purpose is to help teachers and others to predict success or failure on the learning path of a particular student, on a given course, and to take the proper measures to avoid failure. The architecture of SPS is depicted in Figure 6.

![Figure 6: The Student Prediction System Architecture](image)

BUILDING SYSTEM BELIEFS AND STUDENT MODELS FROM MOODLE

In order to be able to produce SMs, the computer system must have some beliefs about the student behaviour, face to the pedagogical resources from a given domain. These beliefs, as we had already seen, are integrated into $B_C$. MOODLE has an activity logger to register users’ accesses (i.e., user ID, IP and time of access) and the activities and resources that have been approached. From the log, MOODLE is able to generate, for each student, activity reports. This information can be combined with biographical, socio-economic and cultural data, as well as former academic history, in order to obtain a more complete representation of the students’ evolution. Figure 7 configures a simplified information model of MOODLE activity log.

![Figure 7: Excerpt from MOODLE Information Model](image)

Generally, the idea is to find patterns, correlations and association rules on data sets of the student interactions with MOODLE. The discovering of association rules is used to come across elements that occur together in data sets, with a given confidence and support values. These values may now be used to establish an order relation on the set of beliefs.

$$B_C = \text{course grade}(S, \text{shoe design}) \leftarrow \text{num course visits}(S, \text{shoe design}, N) \land \text{avg course time}(S, \text{shoe design}, T) \land V \geq 3000$$

Extraction of System Beliefs $B_C$

It is now possible, using data mine techniques for mining association rules from databases, in a way similar to the one that was exploited by Lukichev, Diaconescu and Giurca (2007), which is in itself based on the work of Agrawal, Imielski and Swami(1993), and Agrawal and Srikan (1994), to derive a set of beliefs from MOODLE logs of student transactions, to establish $B_C$.

Program 8: Extract of System Beliefs $B_C$

We use Rule Based Programming (RBP) (Kowalski and Levy, 1996), here given in terms of productions of the Multi-Valued Extended Logic Programming (MVELP) language. In RBP, given a rule in the form $Q \leftarrow P$, $Q$ is triggered whenever $P$ occurs. As an example, we may express association rules in terms of a MVELP program (Program 8). In this example, the system believes:

- That $\text{motivation}$ is a function of the number of visits to the platform, plus the average time per visit;
- That it is required to attend the $\text{drawing}$ course before attend $\text{shoe design}$ course;
- That high number of visits to lesson $\text{shoe cad SW}$ and high average time spent with its pedagogical resources, usually predicts success on that lesson;
That some number of visits to shoe design plus some average course visit time, when above 3000, probably means that the student will grade on that course;
That grade point average is the mean of the grade points of all attended courses.

Systems beliefs can also be added or edited manually by teachers, tutors, psychologists or others (e.g., motivation, course_requirements and grade_PA).

Building and Updating System Models

The SM can be initialized through a combination of system default assumptions and the substance of inquiries presented to the students by the time of their enrolment on a course, being updated through the student interactions with the system. Besides, there is information we can not derive from log files (e.g., school attitude or socio-economical status). Indeed, this information must come from other sources.

\[
SM_{Ana} = \begin{cases} 
\text{had_attended(ana, drawing)} \\
\text{num_visits(ana,20)} \\
\text{avg_visit_time(ana,200)} \\
\text{num_course_visits(ana, shoe_design,16)} \\
\text{avg_course_time(ana, shoe_design,200)} \\
\text{grade_P(ana, drawing,16)} 
\end{cases}
\]

Figure 8: SM for Ana

Let us consider the following subset of MOODLE metrics, for a given student, attending a given course:
- num_visits: number of visits to the Moodle platform;
- avg_visit_time: average time per visit;
- num_course_visits: number of visits to the resources of a course; and
- avg_course_time: average time spent per course visit.

For example, the student Ana, attending a shoe design course, may have the following SM (Figure 8). This logic program stands for the facts that the system knows about the student. The system believes about student behaviour have not yet been applied to the SM of Ana.

In fact, once the student model has been initialized, there are two sources of information on the basis of which the student model may be updated: the student’s inputs to MOODLE, and the current contents of \( B_c \).

\[
SM_{Ana} = \begin{cases} 
\text{had_attended(ana, drawing)} \\
\text{num_visits(ana,20)} \\
\text{avg_visit_time(ana,200)} \\
\text{num_course_visits(ana, shoe_design,16)} \\
\text{avg_course_time(ana, shoe_design,200)} \\
\text{grade_P(ana, drawing,16)} \\
\text{motivation(ana,4000)} \\
\text{course_grade(ana, shoe_design)} \\
\text{grade_PA(ana,16)} 
\end{cases}
\]

Figure 9: SM after applying the System Beliefs, \( B_c \)

Generally, the SMB gets the SM of a given student, analyses the SM, evaluates the quality of SM’s information and updates the SM with its beliefs about the student. These beliefs result from instantiating the system set of beliefs, \( B_c \), to this particular student.

For instance, it is possible to produce these pinpointings for Ana (Figure 9).

We can see that the system updated the SMs with instantiations of its beliefs about motivation, course_grade and grade_PA. These beliefs stand for the information given by SMB.

When the SMB encounters evidence that the current SM is inaccurate, for example, by observing that the student acts differently to the way the SM would predict, namely due to higher number of visits to a course, more time spent with resources, higher percentage of right answers, then the SMB may try to find and adjust those components necessary to enable the model to correspond to the observed student behaviour (this has implicit a time dimension not considered in this work).

STUDENT PREDICTION MODULE

In order to predict the success of the students on a given course, we must learn what the best predictors of success on that course are. In the social sciences, multiple regression procedures are very widely used in research. In our work it is assumed that those predictors (as well as its coefficients) are already known, and are a subset of those used by Kruck and Lending (2003). For example, let us consider the following attributes of the SM that are predictors of success on a course \( C \) (with a certain correlation coefficient \( R \)):

- motivation, \( M \): real
- had_attended, \( H \): \{0 (false), 1 (true)\}
- grade_PA, GPA: real

and the corresponding regression equation, or prediction model:

\[ Y = a + b_1 \times M + b_2 \times H + b_3 \times GPA \]

where \( a \) is the intercept coefficient, \( b \) the slopes or regression coefficients and \( Y \) is the grade point prediction for the student on the course.

SPM gives a prediction of student success on \( C \) by displaying that of a student that attended the course in the past, which is akin to the applicant being evaluated (in education, past behaviour is determined to be the best predictor of future behaviour (Alemoni, 1977)).

Indeed, and in terms of implementation, we draw on Case Based Reasoning (CBR) (Aamodt and Plaza, 1994), whose knowledge base of past cases is given by \( B_c \). Generally, the reasoner (Figure 10) operates in the form:

- Find the set of SMs of students that attended \( C \), before having attended the course. Then retrieve those similar to the SM being analyzed, i.e., \( SM \). Our approach is to use the prediction model itself to compute the similarity. Roughly speaking, the retrieved set is \( \{SM\} \) such as \( Y(SM) \sim Y(SM) \).

Complementarily, the reasoner may also use system beliefs to fine grain the selection.
• Reuse the SMₐ that best matches SMₐ to estimate the probability of success. Label SMₐ with that probability.
• Revise SMₐ. Assess the grade point of SMₐ after attending C: does it confirm the prediction? At this stage, a teacher should decide measures to remedy possible lacks of knowledge.
• Retain the prediction and the measures decided to avoid failure together with SMₐ.

A well defined reasoner must not only solve the case but also explain the solution. The student must be informed and confronted with the predictions and the measures decided for him/her.

Figure 10: Case Based Reasoner (Aamodt and Plaza, 1994)

The framework just presented above is a theoretical model of a SPS. The development is at an early stage. For now, in order to achieve this first implementation, we favour the simplicity of the multiple regression prediction model and that of the reasoning process, upon more complex, though more reliable, prediction models and reasoning processes. After the first version and the analysis of its results, we aim to evolve the system, introducing a more reliable prediction model and a more accurate reasoning process.

CONCLUSIONS

The particular nature of the learning process demands grounded decisions taken on time, to be able to detect problems when they arise and avoid failures. A SPS, as the one described here, is able to build and evolve SMs from the logs of an e-Learning platform. It is also able to output predictions of student learning paths, with a given Quality-of-Information, using CBR and a multiple regression prediction model. The Quality-of-Information is critical to assure the credibility of the whole process of forecast and, therefore, the verdict of the decision makers.

There is much work to be done on the prediction model. Bayesian Networks are better fitted to model the conditional relationships between SM’s predictor attributes. There is also work to do on behalf of improvements on the representation of cases (SM) and on the overall reasoning process. We are currently working on these issues. Finally, we are putting up a prototype of a Student Assessment System (Almeida, 2008) which includes a SPS similar to the one we have just described.

ACKNOWLEDGEMENTS

We would like to thanks the Professional School Centro de Formação Profissional da Indústria de Calçado, São João da Madeira, Portugal, for the kindness in letting us to use its MOODLE e-Learning Platform in this work.

REFERENCES


AUTHOR BIOGRAPHIES

PAULO ALMEIDA is a Systems Engineer and a MSc student at the Department of Informatics, the University of Minho, Braga, Portugal. He has been working for several years on re-engineering Professional Schools’ business processes and technological infrastructures. He has also work done on the fields of systems integration and networks design and administration. He is currently the Chief Information Officer at the Professional High School Centro de Formação Profissional da Indústria de Calçado.

PAULO NOVAIS is Professor of Computer Science at the Department of Informatics at the University of Minho. 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, Multi-agent Systems, Ambient Intelligence and Collaborative Networks.

JOSÉ NEVES is Full Professor of Computer Science at the Department of Informatics 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 Non-monotonic Logics, Knowledge Representation and Reasoning Systems and Learning.
EVALUATION AND CLASSIFICATION TOOLS
A Multiscale Simulation Environment for Performance Evaluation of high reliable heterogeneous Communication Networks

Andreas Lewandowski, Ralf Burda, Sebastian Subik and Christian Wietfeld
Communication Networks Institute (CNI)
Dortmund University of Technology, Germany
email: {andreas.lewandowski|ralf.burda|sebastian.subik|christian.wietfeld}@tu-dortmund.de

KEYWORDS
Simulator Coupling, Performance Evaluation, Wireless Communication Networks, Discrete simulation

ABSTRACT
Mobile wireless communication networks can only be accurately modelled by a bunch of parameter functions due to many uncorrelated influence factors. A Multiscale Network Simulation Environment (MNSE) for an exact effigy of all cross correlations is needed. Where existing Cross-Layer modelling environments mostly respect only one single parameter system, the Multiscale Simulation Kernel based on OMNeT++ (Varga 2001)(Varis 2008) considers the parameter scales Protocol, Radio Channel and User Mobility and provides a generic interface to add other simulation tools. The present implementation combines the protocol simulator OMNeT++, a Radiowave Propagation Simulator, a 3D environment modeler and a Mobility Simulator. On account of an asynchronous coupling architecture, the distribution of simulation load over several simulation environments is possible. The implementation ensures a synchronous data exchange between single components.

INTRODUCTION
In order to generate a holistic model of a wireless communication network (WCN), the dynamic interaction between network infrastructure, environment and especially mobile objects must be considered. Mobile objects are not necessarily subscribers of the network as even a moving passive object may cause interaction with the radio channel.

The modeling of the wide range of effects and influences is mandatory for the design and operation of a wireless communication network (Dhau et al. 2005). The enhancing optimization calls for more detailed modeling, especially taking into account the interaction between system and environment.

Our Multiscale Network Simulation Environment (MNSE), described in this paper, focusses on the implementation of an abstract simulation kernel, where best-in-class simulation tools can be docked on by using standard TCP sockets. This architecture claims a quasi photo realistic model of the total system by taking into account interdependencies between the influencing variables.

Currently, OMNeT++, as a network simulator, is not able to model these dynamic influences accurately but is capable to be extended. Several claims are made by the MNSE modelling approach:

- A 3-D model of the operating environment ensures the model of influences of the environment. The scenario model created by Google Sketch-Up is the base for the other simulation environments.
- A highly accurate Ray Tracing Tool is taken for the calculation of the radio channel properties. This Radio Channel Propagation Simulator (RPS) (Deissner et al. 2005) is an industry standard tool and also used by mobile network operators.
- Realistic model of user mobility is created by using dedicated path and dynamic group mobility models. These mobility models are supported by the MOBILE Object Simulation Environment (MOose) (Michaelis and Wietfeld 2006) developed by CNI.
- Use of a numerical processing tool for calculation and optimization of PHY- and MAC-layer behaviour like bit error rate, adaptive modulation schemes and resulting data rates.
- The full model of dynamics is captured by the Central Event Broker (CEB) implemented in OMNeT++ which manages the major data exchange between the simulation environments. The simulation data is concentrated in a central data management system from where OMNeT++ can request relevant simulation results.

This paper proceeds by explaining the Multiscale Simulation Model assumption and the implemented simulation architecture in detail. A new technique for simulation setup is presented before we give a brief survey of the simulation flow and describe the used simulation coupling schemes. A detailed application scenario demonstrates the powerful compilation of MNSE before this paper is wrapped up by a short conclusion and an outlook to future improvements.
THE MULTISCALE SIMULATION MODEL

The Multiscale Simulation Model can be classified into source models and system models (Kuehn 2006) which are able to interact by using the Central Event Broker (CEB). The source models generate triggers and messages and are also the consumers of the latter. In short, any events which drive on a typical simulation are due to the source models.

A source model represents network participants which can be active or passive and delineates all kind of activity for each object. This includes generated network traffic, user mobility and transmission settings. Especially radio parameters like transmit power and antenna orientation are valid examples for a parameterized source model.

A system model is thought to represent the dynamic interaction between all active source models. At first, the physical models of the network participants, e.g. body model and received power have direct influence on the physical connectivity. These effects also result in a reverse impact on the source entities. Analog, mobility of active or passive objects can cause group effects which influence the mobility and communication behaviour of a single user.

A major challenge of MNSE is the efficient brokerage of events between sources and sinks as well as the coordination of different simulation tools. This task is taken over by the CEB in the overall model. The CEB ensures that events are reaching just the designated set of receivers. In a radio model this is needed in order not to hand over radio transmission events to nodes which would drop the signal anyway due to an insufficient signal level. Moreover, the parallel and yet synchronized execution of different simulation tasks distributed on a server cluster is a hot topic, yet not taken into account for this paper.

SIMULATION ARCHITECTURE

Figure 1 shows the simulation architecture where the central event broker is the core module which manages and interprets inputs and outputs coming from different subcomponents. This approach ensures a synchronous behaviour of all subsidiary components. Synchronisation messages are created on demand, with which single threads can be stopped or started again. The whole system can be controlled with a Simulation Control and Result Processing (SCR) console which ensures the scenario input and coherence between the different modelling bases of the simulation systems. A user interaction component is responsible for the GUI driven capturing of the application scenarios and an online result processing mechanism.

Single simulation systems can be accessed using standardised TCP sockets on which different best–in–class tools can be docked on.

To-date implementation includes the mobility simulator MOOSE, a Radiowave Propagation Simulator (RPS), Matlab for numerical calculation of channel characteristics, OMNeT++ and the included INET framework as shown in Figure 1. Further important components of the MNSE are the individual interface adaptors between each tool and the CEB and the Automated Model Integrator (AMI), based on the Model Information Base (MIB), which provides information about interfaces and file formats. Google Sketch-up is used for the 3D modeling of the environment scenario.
In fact the construction of interfaces between the CEB and simulation tools is a very demanding work. However, as being needed only in case of framework integration for a new tool, the effort is one time only and pays off rather fast during the deployment of MNSE.

Hence the open architecture ensures extensibility in terms of adding or exchanging simulation environments. Different types of interfaces (described later) are used for an optimal coupling of diversified simulation environments.

SIMULATION SETUP

The simulation setup is the critical task for fast engineering in this model approach. Several simulation and modelling environments need synchronous inputs for the initial setup. Especially coordinate transformations, transmitter settings, generation of mobility traces accordingly to the environment and the choice of appropriate OMNeT++ modules are very time consuming.

In order to minimize the input settings overhead, a so called Automatic Model Integrator (AMI) was developed which is able to mediate already done inputs for the outstanding simulation subcomponents.

The process of simulation setup is now achieved in the following steps:

1. Generation of a 3D environment model in Google Sketch-up. The outsourcing of this function from RPS 3D modeler is time saving because of a very intuitive user interface and Google Earth support.

2. The spatial model can then be imported in RPS via an AutoCAD file where material parameters are specified out of a huge material database. Transmitter settings like position, transmit power, operating frequency band and antenna configuration must be specified. A dedicated namespace for transmitter synonyms are chosen to be interpretable for AMI which is then able to specify appropriate network technologies and modules for OMNeT++ and MOOSE.

3. Now the AMI is able to create complete simulation setup files for the protocol and mobility simulation subcomponents.

The setup process is accomplished.

COUPLING STRATEGIES FOR SIMULATION ENVIRONMENTS

Two different types of interfaces have been created, as not all coupled simulation environments require equal data exchange rates. A passive one (dashed lines in Figure 2) and an active one (solid lines in Figure 2) have been constructed and can be used and exchanged on demand.

The passive interface

The passive interface is considered to manage storage and retrieval processes from the central database. A generating simulator (e.g. MOOSE) will generate all information in a pre-run and store the results in the data base from which the CEB will retrieve a snapshot of all positions any time this is needed. This has been implemented by means of an MySQL data base and text-based trace files to achieve a high degree of flexibility.

The 3D modeler is also connected via a passive interface to RPS because the environment model is needed for radio propagation after construction in Google Sketch-up.
The active interface
The active interface is a superset of the passive one in the sense that the data source may update the information in the database while the simulation is executing. As an example, coverage of a particular location in a wireless communication network may change due to fading effects if objects move. If the valid scope of information is just bilateral and temporal, a simple TCP socket connection is established between source and sink by the CEB. Furthermore, socket connections are used by the CEB to trigger update runs of the system model. A special performance issue is the connection to RPS. An integrated COM interface enables the use of RPS without the GUI. A given number of functions are accessible to change the properties of transmitters and the scenario by CEB.

SIMULATION FLOW
The following description demonstrates the dynamic interaction between simulation environments in a simulation flow graph as shown in Figure 3.

A 3D model of the environment is generated by Google Sketch-up and transferred to Automatic Model Integrator where an initial scenario setup for RPS, MOOSE and OMNeT++ is generated. The source models must be initialized in this first phase because position and radio settings have influence on the initial coverage estimation.

That is why the initialization phase accomplishes the calculation of mobility traces and radio channel in parallel. The data can then be stored in the database, as MOOSE relies on a passive interface in this stage of expansion. In future releases it is planned that mobility can be influenced by the radio channel or the protocol e.g. if the received power is very low a person would go back to the position where the connectivity is better.

The protocol simulation can be started afterwards in order to grant access on relevant mobility and radio channel data.

The dynamic interaction between source and system model is viewable during the simulation execution. Whilst simulation, the influence of moving objects on the connectivity map is taken into account by a dynamic new calculation of the radio channel. The actual object constellations are assembled by the CEB and stored in the database accessible for RPS. The integrated decision support function in the CEB decides whether the influence of moving objects on the radio channel is as big as a new calculation is meaningful.

The current object constellation is then taken by RPS for the new calculation. While RPS calculates the new radio coverage, OMNeT++ is able to proceed the simulation as far as possible. An integrated service thread handles the TCP connections to the external simulation environments.

ITERATIVE PERFORMANCE EVALUATION UP TO FULL MULTISCALE FUNCTIONALITY
Time measurements of total simulation execution time are taken for a performance comparison between different expansion stages of the simulation environment. A simple WiFi scenario as shown in figure 4 was taken as reference and is characterized as follows:

- The number of nodes is increased incrementally.
- Each node executes a simple Pinging Application.
- The scenario is formed by a simple 4 · 6m room.
- A simulation time of 1 minute is examined.

Additional simulation environments have been added in four steps to reach the full multiscale functionality in order to show the impact of each tool on the performance of the whole simulation environment.

Figure 4: Iterative Implementation of RPS Radio Channel in Protocol Simulation for Performance Evaluation

1. The WiFi scenario is calculated in OMNeT++.
   Moving hosts are pinging the access point.
2. Mobility traces are calculated by MOOSE and queried out of a trace file.
3. A passive RPS interface is established where radio channel information is queried out of a MySQL database. The radio channel is not updated during the simulation.

4. Full multiscale functionality with an active RPS interface using intelligent decision support and point to point calculations for channel updates. Moving objects are updated in RPS with an object constellation file created by Central Event Broker.

Figure 5: Simulation Time Analysis

The simple OMNeT++ simulation is the benchmark for performance comparison. As depicted in Figure 5, each extension with additional subcomponents rises the processing time for the simulation execution.

The integration of the MOOSE mobility simulator (OMNeT+MOOSE) needs additional queries of a mobility trace file, but the rise of the execution time is negligible.

It can be seen that the passive multiscale approach (OMNeT+MOOSE+RPS passive) needs again a bit more execution time because of database queries for radio channel data done here on every movement step. The overall performance is still very good and comparable to the simple OMNeT implementation.

The active RPS interface (OMNeT+MOOSE+RPS active) is more accurate and does not rely on the assumptions made for passive interface concerning the radio channel. It shows a higher execution time because the radio channel is recalculated online for all network participants during the simulation. Exact declarations about interference, bit error rate and adaptive retransmissions can be achieved by a postprocessing of RPS results in Matlab. This high channel accuracy is especially needed in heterogeneous network setups and indoor scenarios for example evaluation of indoor localization protocols which rely on the radio channel (Burd et al. 2008).

By applying a simple decision support function to the radio channel update process, higher processing efficiency can be achieved. For instance, not the whole scenario must be updated in case of a packet sent event. Just local influences must be recalculated depending on objects’ position change events.

APPLICATION SCENARIO: High reliable heterogeneous communication network for rescue operations

In order to demonstrate the powerful compilation of MNSE, the development of a simulation scenario for a heterogeneous communication network for high reliable fire fighter communication is shown in the following chapter.

Today’s fire brigades rely on traditional communication and visualization techniques. In order to enhance the communication and efficiency in fire fighter missions, we are developing an innovative communication approach in the project Galileo4FireBrigades. The explanation of the concept is done by the simulation setup of a forest fire scenario shown in Figure 6.

A 3D model delivers detailed information about the place of action in this scenario-based approach. Fire engines arrive, fire fighters move out to the origin of
fire, coordination instances receive information about occurring events and the water supply must be granted. For basic communication concerns we propose to use group communication based on WiFi, as this technology operates independent on public telecommunication services like GSM or UMTS and can be used autaric. The radio propagation evaluation in figure 6 shows radio shadows e.g. in the forest, which can be balanced with so called mobile WiFi Dropped Units (Wolff et al. 2008).

The mobility behaviour of the rescue forces seems to be intuitive. Highest densities are at the fire front as this process is a line event. This exact mobility model combining predefined areas and dedicated path mobility can be achieved by MOOSE.

The fire fighter surveillance aspect of our proposed communication network is visible in Figure 6c. Vital and environment sensors like pulse, temperature (environment and body), gas and position are transmitted utilizing IEEE802.15.4/ZigBee networks to the group leader’s PDA, which then forwards relevant sensor information to the coordination instance of the fire brigade.

As we have already seen, in some cases, occurring radio shadows cannot be balanced by Dropped Units. The prospective enhanced Galileo Search-and-Rescue Service is then used for out of coverage communication. In (Lewandowski et al. 2008) we have already shown using MNSE, that direct communication – especially in emergency situations – is possible in near real time with the coordinating instance of the fire brigades. Figure 6c shows the exact satellite constellation calculated by OMNeT++ relying on the Simplified General Perturbations (Kelso et al. 1988) algorithm developed by NASA.

CONCLUSION AND OUTLOOK

This paper demonstrates the powerful compilation of the multiscale simulation approach by the given insight to architecture, simulation flow and performance aspects. Considering the dynamic interaction between scenario, radio channel and moving objects create a realistic model of a wireless communication network. The Central Event Broker architecture ensures a flexible and efficient simulation accomplishment and easy extension and exchange of best-in-class simulation tools.

Especially the performance evaluation of the full multiscale simulation environments leaves room for improvements. Intelligent decision support functions have to be developed to minimize the calculation overhead.

A real system demonstrator for the proposed communication infrastructure will be constructed out of the gained results from the use case scenario. We will then cross validate the real system with the multiscale simulation model.

ACKNOWLEDGEMENT

This work was funded by German Federal Ministry of Economics and Technology (BMWI). Project: Galileo4FireBrigades – Employment of Galileo services for wide area fire fighter missions – 50NA0724

REFERENCES


SERVICE LEVEL AGREEMENT SPECIFICATION, COMPLIANCE PREDICTION AND MONITORING WITH PERFORMANCE TREES

N.J. Dingle       W.J. Knottenbelt       L. Wang
Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ
email: {njd200, wjk, lw205}@doc.ic.ac.uk

ABSTRACT

Service Level Agreements (SLAs) are widely used throughout industry but suffer from specification ambiguities and difficulties in predicting and monitoring compliance. To address these issues, we propose the use of the Performance Tree formalism for the specification and monitoring of Service Level Agreements (SLAs). Specifically, we show how the basic Performance Tree formalism can be adapted to provide a rigorous yet accessible and expressible means to specify common SLA metrics. Using established performance analysis tools that support Performance Trees, this allows system designers to check SLA compliance on formal models of their system before implementation. We also propose an architecture for a system of measurement agents that enables the same performance requirements to be monitored in the context of a live implementation.

KEYWORDS
Service Level Agreements; Performance Trees; Performance analysis; Quality of Service

INTRODUCTION

Many organisations depend heavily on the availability, reliability and performance of key services delivered by internal business units and external organisations. To ensure an adequate quality of service, it is common practice to contractually define the parameters of service provision in the form of a Service Level Agreement (SLA). SLAs specify the type and quality of service to be provided by a supplier in return for a fee paid by a user, as well as any compensation due to the user in the case of sub-standard service delivery by the supplier.

This paper addresses a number of key challenges relating to SLA set-up, compliance prediction using a system model, and monitoring of a live system.

Firstly, it is important to specify SLA metrics in a way that is rigorous and unambiguous yet readily accessible to both supplier and user. Natural language is an obvious and accessible way to specify such metrics, but is neither rigorous nor unambiguous.

Mathematical formalisms such as stochastic logics [1, 2] are unambiguous and rigorous but are not accessible; they are also restricted in the range of concepts they can express. Here we propose the use of Performance Trees (PTs) – a recent formalism for the graphical specification of complex performance queries [17, 18] – for SLA metric specification. By extending the basic Performance Tree formalism using its macro feature, we show how PTs are able to provide rigorous yet accessible metric specification without sacrificing expressiveness.

Secondly, it is often the case that service suppliers need to predict SLA compliance for systems that have not yet been implemented. Such design-time intervention helps to avoid the situation whereby a supplier finds that the system they have built does not meet – and cannot feasibly be adapted to meet – agreed SLA requirements. It is also often the case that service suppliers need to predict the effect on SLA compliance of proposed changes to currently operational systems. In both cases, the construction and analysis of a stochastic model provides suppliers with a low-cost means to make the necessary predictions. Indeed, we show how, thanks to recent support implemented in the PIPE tool [3], the compliance of SLA requirements expressed as Performance Trees can be directly evaluated on stochastic models.

Finally, monitoring of SLA compliance in operational systems poses additional challenges in terms of collecting and processing measurement data in a way that ensures the accurate computation of a given SLA metric. To this end, we present an architecture for a system of monitoring agents that can be used for the run-time evaluation of SLA requirements expressed as Performance Trees.

A wide range of separate studies have been carried out in each of the areas of SLA specification, compliance prediction and monitoring. Regarding SLA specification, investigations about formalising SLA frameworks have been carried out on various types of IT services, for example databases, e-commerce systems and technical support operations [7, 10, 11, 14, 20]. Bouman [4] has pointed out the existing problems with present SLA specifications mechanisms in terms of ambiguity, incompleteness and inefficiency, and at the same time suggested general principles in guiding customers and service providers in specifying SLA requirements. More formally, an XML language SLAng [9] based on the Unified Modelling Language (UML) system model has been introduced that provides a rigorous and unambiguous approach for SLA specification. However, it is rather syntactically complicated for non-IT professionals.

Much research has also been carried out on SLA monitoring for a wide range of IT systems [8, 12, 13, 15, 16]. For example, Pereira [15] describes a hierarchical architecture for monitoring quality of service (QoS) parameters to help the
enforcement of SLAs between a provider and users. Padgett [14] combines SLA specification and compliance prediction by suggesting a set of service metrics and demonstrating their use in predicting CPU usage in a mathematical model. However, the SLA metrics are limited to only CPU usage, and monitoring of SLA compliance is not considered.

To the best of our knowledge, therefore, the work presented in the present paper is the first time a unified environment has been proposed for SLA specification, compliance prediction and monitoring.

The remainder of this paper is organised as follows. In the next section, we discuss relevant background including SLA metrics, performance analysis techniques and the Performance Tree formalism. In the following section we introduce a Voting System model that is used as a running example throughout the rest of the paper. We then show how Performance Trees can be used to specify various common SLA metrics, including availability, mean time between failures and mean time to repair, response time percentiles, resource utilizations, throughputs and system productivity. Finally, we present our architecture for an online PT-based SLA monitoring system before concluding and discussing opportunities for further work.

BACKGROUND

Service Level Agreement metrics

In this paper we concentrate on metrics related to availability, response times, resource utilizations, throughputs and system productivity. This ensures that we are able to support a superset of the metric-related concepts covered by the QoS specification language QML [6]. We define these concepts as follows:

- **Availability** is the proportion of time during which service is provided to users. Availability is in turn dependent on two further metrics: **Mean Time Between Failures (MTBF)** and **Mean Time To Repair (MTTR)**.

- **Response time** is the time from a user sending a request to receiving a response or the time from job submission to job completion. SLAs are often concerned with means, variances, and percentiles of response times. For example it may be required of postal service that “90% of first class post is delivered within one working day of posting”.

- **Resource utilisation** is the proportion of time for which a given resource is used by a given service. For example it may be required that a service does not utilise more than 10% of available network bandwidth.

- **Throughput** is the average rate at which a given set of activities occurs. For example, it may be required that a system processes a minimum of 3,000 transactions per second.

- **Productivity** is a weighted sum of the throughput of a number of activities, where the weights are user-specified rewards associated with completion of each activity. If the unit of the reward is financial, this measures a system’s **profitability**.

Performance Analysis

Performance is a vital consideration for system designers and engineers. Indeed, a system which fails to meet its performance requirements can be as ineffectual as one which fails to meet its correctness requirements. Ideally, it should be possible to determine whether or not this will be the case at design time. This can be achieved through the construction and analysis of a performance model of the system in question, using formalisms such as queueing networks, stochastic Petri nets and stochastic process algebras.

Having created a stochastic model of the system, it needs to be decided what performance measures are of interest. It is possible to capture such requirements in logical formulae using a language such as Continuous Stochastic Logic (CSL) [1, 2]. These languages provide a concise and rigorous way to pose performance questions and allow for the composition of simple queries into more complex ones. Such logics can be somewhat daunting for non-expert users, and there still remains the problem of correctly converting informally-specified requirements into logical formulae.

Performance Trees

Performance Trees are an intuitive graphical formalism for the quantification and verification of performance properties. They were proposed to overcome the problems associated with logical stochastic property specification highlighted in the previous section. They combine the ability to specify performance requirements, i.e. queries aiming to determine whether particular properties hold on system models, with the ability to extract performance measures, i.e. quantifiable performance metrics of interest.

The concepts expressible in Performance Tree queries are intended to be familiar to engineers and include steady-state and passage time distribution and densities, their moments, transition firing rates, convolutions and arithmetic operations. An important concern during the development of Performance Trees was ease of use, resulting in a formalism that can be straightforwardly visualised and manipulated as hierarchical tree structures.

A Performance Tree query is represented as a tree structure consisting of nodes and interconnecting arcs. Nodes can have two kinds of roles: operation nodes represent performance-related functions, such as the calculation of a passage time density, while value nodes represent basic concepts such as a set of states, an action, or simply numerical or Boolean constants. Table 1 presents a summary of Performance Tree nodes used in this paper.

The formalism also supports macros, which allow new concepts to be created with the use of existing operators, and an
abstract state-set specification mechanism to enable the user to specify groups of states relevant to a performance measure in terms of the corresponding high-level model.

Performance Trees have been integrated into the Platform Independent Petri net Editor (PIPE), thus allowing users to design Generalised Stochastic Petri Net (GSPN) models and to specify relevant performance queries within a unified environment. PIPE communicates with an Analysis Server which employs a number of (potentially parallel and distributed) analysis tools to calculate performance measure [5]. These include steady-state measures, passage time densities and quantiles, and transient state distributions.

To offer greater ease for constructing performance queries, we have recently developed an alternative query construction mechanism called the Natural Language Query Builder (NLQB) [19]. The NLQB guides users in the construction of performance queries in an iterative fashion, presenting at each step a range of natural language alternatives that are appropriate in the query context.

**RUNNING CASE STUDY**

Fig. 1 shows a GSPN model of an electronic voting system\(^1\) which will be used throughout this paper. In the model there are several voters, \(CC\), a limited number of polling booths, \(MM\), and a smaller number of central servers, \(NN\). Voters vote asynchronously, moving from place \(p_0\) to \(p_1\) as they do

\(^1\)http://www.doc.ic.ac.uk/~njd200/voting.mod

so. Polling booths which receive their votes transit \(t_1\) from place \(p_2\) to place \(p_3\). At \(t_3\), the vote is registered with as many servers as are currently operational in \(p_6\).

The system is considered to be in a failure mode if either all the polling booths have failed or all the central servers have failed. If either of these complete failures occurs, then a high priority repair is performed that resets the failed units to a fully operational state. If some (but not all) of the booths or servers fail, they attempt self-recovery via transitions \(t_4\) and \(t_5\) respectively. The system is considered to be available and will continue to function as long as at least one polling booth and one server remain operational.

To facilitate the reasoning about failures and repairs, we augment the state vector of the system model with two boolean
components: just-repaired which is set to true by any transition which moves from a failed to an available state, and just-failed which is set to true by any transition which moves from an available to a failed state.

Note that the rate at which failures (recoveries) occur depends on the number of currently operational (failed) units and so the transitions $t_2$, $t_3$, $t_5$ and $t_6$ are modelled with marking-dependent rates. Also, the rate at which voters vote depends on the current number of voters and available polling booths, and the rate at which polling booths register cast votes depends on the number of available servers, and so $t_1$ and $t_3$ are also modelled with marking-dependent rates.

The system we study features 100 voters, 10 polling booths and 10 servers, and its underlying Continuous Time Markov Chain (CTMC) contains 93,274 states.

SLA SPECIFICATION WITH PERFORMANCE TREES

To demonstrate the use of Performance Trees to describe and predict compliance with SLA requirements, we have constructed some common SLA metrics using parameterised macros and evaluated them for the Voting System model. Our methodology is not limited to those metrics described in this section, however, as the extensible nature of Performance Trees allows the user to construct macros for the metrics which are most relevant to them.

Availability

![Availability macro definition](image)

**Figure 2: Availability macro definition**

Fig. 2 shows the PT macro definition for availability. This takes as input a parameter, $X$, that is a state function that returns 1 if the system is available in the state, and 0 otherwise. Evaluating the SSS operator yields a probability mass function (pmf), the domain of which is 0 and 1 (corresponding to system non-availability and availability respectively) and the range of which is the steady-state probability of each domain value. Computing the expected value (via the first moment) of the pmf yields the steady-state probability of finding the system in an available state.

Suppose the SLA for the Voting System specifies that it shall have an availability greater than 99%. The corresponding Performance Tree definition of this requirement, which makes use of the PT macro defined above, is given in Fig. 3. According to the model description the system is considered to be available if at least one server and one polling booth are operating. The corresponding state function therefore evaluates to 1 if $\#(\text{servers}) > 0 \land \#(\text{booths}) > 0$, and to 0 otherwise. By evaluating this query on the model (using PIPE's Performance Tree Query Editor and associated evaluation environment) we find that the system is predicted to achieve 99.88% (to 2 d.p.) availability and therefore to meet its SLA requirement.

Mean Time Between Failures

![Mean Time Between Failures macro definition](image)

**Figure 4: Mean Time Between Failures macro definition**

Another important measure is the Mean Time Between Failures (MTBF), for which the corresponding Performance Tree macro definition is shown in Fig. 4. This requires two sets of states as input: the states where the system has just been repaired, $R$, and the failure states, $F$. These states can be defined using the evaluator’s built-in state set definition tool. Evaluating the PTD operator yields the pdf of the first passage time from state set $R$ to state set $F$. Computing the expected value (via the first moment) of the pdf yields the MTBF.
Fig. 5 shows the instantiation of this macro for an SLA requirement that the MTBF of the Voting System is greater than 1000 time units. From the model description above we have that the system is considered to have failed if either all servers or all booths have failed. By evaluating this query on the model we find that the MTBF is 661.69 time units (to 2 d.p.) and we can therefore conclude that the system will not meet its MTBF requirement.

**Mean Time To Repair**

![Image of Mean Time to Repair](image)

**Mean Time To Repair** (MTTR) indicates how quickly a system recovers from a failure state. As shown in Fig. 6, it requires two set of states as input: the states where the system has just failed, $F$, and the states where the system is repaired, $R$.

Fig. 7 shows the instantiation of this macro for an SLA requirement that the MTTR of the Voting System is less than 5 time units. By evaluating this query on the model we find that the MTTR is 0.96 time units (to 2 d.p.) and we can therefore conclude that the system will meet its MTTR requirement.

We note that an alternative definition of availability is:

$$\frac{MTBF}{MTBF + MTTR}$$

Substituting in the above values for MTBF and MTTR, we have that the Voting System’s availability is 99.86% (to 2 d.p.), which agrees very well with the value calculated above from the mean of the distribution of the corresponding state function.

**Response time percentiles**

![Image of Response time percentile](image)

Fig. 8 shows the Performance Tree definition of a response time percentile requirement while Fig. 9 shows the Performance Tree corresponding to the SLA requirement that “the system shall be capable of processing all voters within 30 time units 95% of time”. The graph in Fig. 10 shows the cumulative distribution function of the time taken to process all voters with the 95% confidence interval at $t = 25.5$ time units marked. As the SLA requires this time to be less than 30 time units, we can conclude that the system will meet this response time percentile requirement.

**Resource utilisation**

Fig. 11 shows the Performance Tree macro definition for an SLA resource utilisation requirement. This is similar to the Availability macro, save for the fact that its input state fun-
tion describes when a particular resource is said to be busy rather than when the system is available.

Fig. 12 poses the question for the Voting System “For what proportion of time is at least one polling booth in use?”. By evaluating this query on the model we find that the proportion of time for which this is the case is 99.93% (to 2 d.p.).

**Throughput**

Throughput is the measure of the rate at which certain activities occur, for example the amount of data transferred over a communications link per second. It is also a fundamental element for evaluating the productivity of a system. In Performance Tree terms, throughput is described as the mean occurrence rate of actions in the corresponding system model. The corresponding macro simply consists of one Firing Rate node.

Fig. 13 shows the Performance Tree query asking for the average number of votes being cast per time unit. By evaluating this query we find that the Voting System achieves a throughput of 0.075 voters per time unit (to 3 d.p.).

**Productivity/Profitability**

Based on the definition of throughput, Performance Trees can be used to provide a measurement of the productivity of a system. As shown in Fig. 14, for an SLA productivity can be defined as a sum of the products of the mean occurrence rate of certain actions and their corresponding impulse rewards, which can be positive or negative.

Fig. 15 shows the Performance Tree of the SLA requirement that the Voting System “shall make a profit of more than 20 currency units per time unit, based on earning 2 currency units from each vote successfully cast but paying 100 currency units for each high-priority server repair and 50 currency units for each high-priority polling booth repair”. In this example, the first action/reward pair is made up of the rate at which votes are cast and the earnings from one vote; the second pair consists of the rate of high-priority server repairs and their cost, and the third pair consists of the rate of high-priority polling booths repairs and their cost.

Calculating the throughputs of the three relevant transitions and multiplying by the rewards for each reveals that the total profitability of the system is -0.02 currency units per time unit (to 2 d.p.). From this, we can conclude that the system does not meet its SLA requirement for profitability and indeed is predicted to cost more to run than it generates in revenue.
ARCHITECTURE FOR ONLINE SLA MONITORING

In this section we propose a unified architecture to enable PTs to be used to monitor SLAs in a real system as well as to evaluate SLA requirements on a system model. This will provide users with an accurate and accessible approach to evaluate actual system performance against the same SLA specifications for which the system model has been analysed. The general structure of the architecture is shown in Fig. 16. To gather data on the system’s behaviour, client-side event logging agents are installed that collect performance data relevant to the metrics specified in the PT operator nodes of the corresponding SLA definition. For example, if the SLA contains a passage time density operator, such agents would record the time taken for each passage of the system from the start to target states. Once enough passages have been observed it is possible to construct an approximation of the density using a histogram. The data thus collected can be then stored in an online database that is accessed through an intuitive user interface which permits users to verify the actual system against SLAs using the same PT’s SLA specification interface as for model analysis. This is done by constructing their SLA requirement query and submitting it to the SLA monitor. The SLA monitor extracts the relevant data from the online database according to the request received and generates a compliance report. This architecture can also be used to verify the system model against the real implementation to check if it correctly reflects the behaviour of the live system. This will allow investigation of new system configurations to be conducted to determine their likely effectiveness. For example, users could investigate adding an extra server or increasing the bandwidth to their system to predict the likely performance improvement. They could then decide whether or not the improved performance warranted the expense of implementing such changes.

CONCLUSION

In this paper we have demonstrated how Performance Trees can be used to reason about SLAs. Using macro functionality, we have presented PT-based definitions for common SLA concepts such as availability and productivity, and we have demonstrated the analysis of such metrics using a case study of a Voting system. We have also proposed an architecture based on the use of monitoring agents to collect information about the performance of the system once implemented. This can then be used to monitor compliance with an SLA using the same Performance Tree framework employed for pre-implementation analysis.
In the future we intend to extend the work presented here in a number of ways. We will add support for the expression of SLA metrics in natural language by augmenting our existing Natural Language Query Builder; this will make the construction of SLA-specific Performance Trees even more intuitive. We will also implement the on-line monitoring architecture and demonstrate its applicability to a real-world example.

REFERENCES


SIMULATION AND MODELLING OF RAID 0 SYSTEM PERFORMANCE

F. Wan  N.J. Dingle  W.J. Knottenbelt  A.S. Lebrecht
Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ
email: {yffw05, njd200, wjk, asl102}@doc.ic.ac.uk

ABSTRACT

RAID systems are fundamental components of modern storage infrastructures. It is therefore important to model their performance effectively. This paper describes a simulation model which predicts the cumulative distribution function of I/O request response time in a RAID 0 system consisting of homogeneous zoned disk drives. The model is constructed in a bottom-up manner, starting by abstracting a single disk drive as an M/G/1 queue. This is then extended to model a RAID 0 system using a split-merge queueing network. Simulation results of I/O request response time for RAID 0 systems with various numbers of disks are computed and compared against device measurements.

KEYWORDS
Queueing models; Simulation models; Storage system performance

INTRODUCTION

The Redundant Array of Inexpensive Disks (RAID) [8] is a storage technology which can ease the widening performance gap between the processor and I/O subsystem. By spreading I/O operations over multiple disks, the bandwidths of several hard disks can be utilised and overall I/O throughput can be increased. Once designed for high-end servers and mainframes, products with RAID functionality are increasingly popular and today can even be found in chipsets for desktop computers, for example the Intel 915P chipset for Intel Pentium 4 processors [3]. Given the widespread adoption of RAID systems and the fact that the I/O subsystem is often a performance bottleneck, it is important to be able to predict RAID system performance.

In this paper we present a simulator that predicts the performance of a RAID level 0 (i.e. striping without redundancy) system. We work in a bottom-up manner, basing our single disk simulation on the analytical model for a zoned disk drive presented in [7]. Since this model abstracts a disk drive as an M/G/1 queue, we construct a validated M/G/1 queuing simulator to implement it. We further abstract a RAID system as a split-merge queueing network [4], and develop a corresponding simulator which outputs a cumulative distribution function (cdf) of the I/O response time for a specified arrival rate, size of request and request type (read or write). To validate our simulator, we compare cdfs of the simulations and measurements.

SINGLE DISK MODEL

Before we can model the performance of a RAID system, we must be able to model the performance of its constituent disk drives. We therefore begin with the construction of an effective single disk simulator, implementing an established analytical model for a zoned disk drive. Zoning arises on modern hard drives because there are more sectors on cylinders on the outside of the platter than those closer to the centre. A zone is a contiguous collection of cylinders which have the same number of sectors and thus the same storage capacity. Since disks rotate with constant angular velocity, data throughput is higher for outer zones than for inner ones [7]. We model a disk drive as an M/G/1 queue, in which the service time is defined as the sum of seek time, $S$, rotational latency, $R$, and $k$-block transfer time, $T_k$.

Seek Time

Seek time, $S$, is the time taken to move the actuator arm to the track where the destination sector lies. This is a function of $D$, the distance between the starting and target cylinders [2]:

$$S(D) = \begin{cases} 0 & \text{if } D = 0 \\ a + b\sqrt{D} & \text{otherwise} \end{cases}$$

where $a$ and $b$ are constants defined in terms of the disk geometry:

$$a = \frac{\text{minseek}\sqrt{\text{Cyls}} - \frac{1}{\text{maxseek}}}{\sqrt{\text{Cyls}} - \frac{1}{1}}$$

$$b = \frac{\text{maxseek} - \text{minseek}}{\sqrt{\text{Cyls}} - \frac{1}{1}}$$

Here $\text{Cyls}$ is the total number of cylinders on the disk, $\text{minseek}$ is the track-to-track seek time and $\text{maxseek}$ is the full-stroke seek time. We note that $\text{minseek}$ and $\text{maxseek}$ are potentially dependent on whether the I/O operation is a read or a write.

Assuming I/O accesses are uniformly randomly distributed across disk sectors, the seek time cdf, $F_S(t)$, can be defined in terms of the seek distance cdf, $F_D(t)$ [2]:

$$F_S(t) = F_D \left( \frac{t - a}{b} \right)^2$$

The probability density function of $D$, $f_D(x)$ is calculated in [10] as:

$$f_D(x) = A + Gx + Ex^3 \quad (0 \leq x \leq C - 1)$$
The constants are defined as follows:

\[ A = \frac{V(Cyls - 1)}{3\gamma^2} \]
\[ G = \frac{-V + \beta^2(Cyls - 1)^2}{3\gamma^2} \]
\[ E = \frac{\beta^2}{3\gamma^2} \]
\[ V = 6\alpha^2 + 6\alpha\beta(Cyls - 1) + 2\beta^2(Cyls - 1)^2 \]
\[ \alpha = \frac{SEC[0]}{spb} \]
\[ \beta = \frac{SEC[Cyls - 1] - SEC[0]}{(Cyls - 1) spb} \]
\[ \gamma = \alpha(Cyls - 1) + \frac{\beta^2}{2}(Cyls - 1)^2 \]

SEC[0] and SEC[Cyls - 1] are the number of sectors on the innermost and outermost tracks respectively, and spb is the number of sectors per block.

**Rotational Latency**

Rotational latency, \( R \), is the time taken for the disk to rotate until the required sector is under the read/write head. It is uniformly distributed between 0 and the time for a disk to make a full rotation \( (R_{max}) \) [2].

**Data Transfer Time**

Data transfer time, \( T_k \), is the time taken to transfer \( k \) data blocks to or from the read/write head. The function to calculate \( T_k \) for cylinder \( x \) of a zoned disk is [10]:

\[ T_k(x) = \frac{k spb R_{max}}{\alpha + \beta x} \]

The cdf of the data transfer time, \( F_{T_k}(t) \) can then be derived as [7]:

\[ F_{T_k}(t) = \left\{ \begin{array}{ll}
0 & \text{if } t < k spb t_{min} \\
\frac{1}{2(t_{max} - t_{min})^2} \left[p + \frac{q}{t} + \frac{r}{t^2}\right] & \text{if } k spb t_{min} \leq t \leq k spb t_{max} \\
1 & \text{otherwise}
\end{array} \right. \]

with

\[ p = (Cyls - 1)t_{max}(2(t_{max} - t_{min})\alpha + (Cyls - 1)(t_{max} - 2t_{min})\beta) \]
\[ q = ((Cyls - 1)t_{max}(-2 k spb(t_{max} - t_{min})t_{min})\alpha + (Cyls - 1)k spb t_{min}t_{max}\beta + (1 - Cyls)k spb(t_{max} - 2t_{min})t_{min}\beta) \]
\[ r = (1 - Cyls)(Cyls - 1)k^2 spb^2 t_{max}^2 t_{min}^2 \beta \]

\( t_{min} \) and \( t_{max} \) are the times to transfer to a single sector on the outermost and innermost tracks respectively.

**Table 1: Seagate ST3500630NS drive parameters**

<table>
<thead>
<tr>
<th>Capacity</th>
<th>500GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of cylinders</td>
<td>60,801</td>
</tr>
<tr>
<td>RPM</td>
<td>7,200</td>
</tr>
<tr>
<td>Sector size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Sectors per block</td>
<td>256</td>
</tr>
<tr>
<td>Time to write a single physical sector on the innermost track ( t_{max} )</td>
<td>0.012064 ms</td>
</tr>
<tr>
<td>Time to write a single physical sector on the outermost track ( t_{min} )</td>
<td>0.005976 ms</td>
</tr>
<tr>
<td>Track-to-track seek time (Read)</td>
<td>0.8 ms</td>
</tr>
<tr>
<td>Full-stroke seek time (Read)</td>
<td>17 ms</td>
</tr>
<tr>
<td>Track-to-track seek time (Write)</td>
<td>1 ms</td>
</tr>
<tr>
<td>Full-stroke seek time (Write)</td>
<td>18 ms</td>
</tr>
</tbody>
</table>

**The Single Disk Simulator**

Our single disk simulator is a simulation of an M/G/1 queue and is implemented using the JINQS Java-based queuing network library [5]. Interarrival times for I/O requests are sampled from an exponential distribution with rate parameter \( \lambda \). The service time for each request is generated by summing samples from the random variables \( S, R \) and \( T_k \) (for some fixed \( k \)). The simulator processes a specified number of I/O requests (usually 5,000) and outputs the response time for each request. These are then used to generate the cumulative distribution function of I/O request response time, as well as other summary statistics such as the mean, sample standard deviation and median.

The operation of the M/G/1 queue simulator was validated for a simple Erlang service time distribution by comparing the cdf generated by our simulator with a known analytical result. In particular, the Laplace transform of the response time density of an M/G/1 queue can be derived using the Pollaczek-Khintchine transform equation [6], and then numerically inverted using Euler inversion [1] to yield the response time cdf. Agreement between simulated and numerical cdfs was found to be excellent, giving us confidence in the correct operation of our simulator.

To generate samples for disk service time, the simulator extracts samples from the analytical cdfs for \( F_S(t) \) and \( F_{T_k}(t) \). Specifically, we sample from a cdf \( F(t) \) by computing the value of \( t \) for which \( U = F(t) \), where \( U \) is a uniformly distributed random variable, \( 0 \leq U \leq 1 \). Since rotational latency is uniformly distributed, a random sample is \( UR_{max} \). The overall disk service time is the sum of the samples for seek time, rotational latency and transfer time.

**Comparing with Measurements**

The output of our single disk simulator was compared with the measurements from a Seagate ST3500630NS disk drive [9]. Table 1 gives the model parameters for this drive. Tests were conducted with an arrival rate of I/O requests \( (\lambda) \) of 0.01 and 0.02 requests/ms with workloads consisting of
$k = 1, 2, 5, 10, 20, 40$ 128KB blocks. For each value of $k$ we ran two tests – one for reads and one for writes. Our single disk simulation matches device measurements very well, especially for workloads consisting of a lower arrival rate and smaller request sizes. We note, however, that the simulated write results do tend to slightly underestimate the measured results – see for example Fig. 1.

When $\lambda = 0.02$ requests/ms and $k > 10$ the simulation results begin to underestimate the device measurements for both reads and writes. For $k \geq 20$ at both arrival rates, the model fails to give accurate predictions of response time because the requests are arriving faster than they are being served, and the queueing theory upon which our simulator is based requires a stable state to function properly.

### RAID 0 MODEL

#### Split-Merge Queues

We now extend the single disk model to a RAID 0 model using a split-merge queueing network [4]. In a split-merge queue with $N$ servers, a job splits into $N$ subtasks which are serviced in parallel. Only when all the subtasks finish servicing and rejoin can the next job split into subtasks and start servicing. Therefore, for each request, the response time is defined as the maximum of the subtasks’ response times.

![Diagram of a split-merge queueing network](image)

The split-merge queueing model fits well with the way in which a RAID system operates, since disk drives in the RAID system can be treated as servers in the queueing network. Users issue I/O requests to the RAID controller which holds arriving requests in a queue. The RAID controller repeatedly dequeues requests and splits them into several sub-requests, each to be serviced by one of the disk drives in parallel. After all sub-requests of a request have been serviced, they are merged and the I/O request completes. We note that newer RAID 0 systems have buffering queues before and after the disks, and so disks can serve the sub-requests of new requests continuously without waiting for earlier requests to complete. Our model will slightly overestimate I/O request response time for such systems.

#### Modifying the Single Disk Simulator

We constructed a RAID 0 simulator by building a split-merge layer on top of our existing single disk simulator. When the RAID 0 simulator receives an I/O request from its queue, it splits the request into smaller sub-requests, and calls the single disk models to calculate the service times of those sub-requests. The RAID simulator selects the maximum service time returned by the single disk models to be the service time of that I/O request.

![Diagram of RAID 0 simulator](image)

Upon receiving a $k$-block read or write request, the RAID simulator (Raid0 hereafter) starts to allocate the blocks to different instances of single disk models (SingleDisk hereafter). The allocation scheme is illustrated in Fig. 3. Each SingleDisk object, $sd[i]$, will read or write a minimum of $\lceil \frac{k}{n} \rceil$ blocks. Additional blocks, $X$, will be added as follows:

$$ X = \begin{cases} 1 & \text{if } i < (k \mod n) \\ 0 & \text{otherwise} \end{cases} $$

For example, if a 7-block write request, write(A, B, C, D, E, F, G), is issued to a 3-disk RAID 0 system, then the 7 blocks will be written to the disks in the following way:
sd[0] write (A, D, G) (k = 3)
sd[1] write (B, E) (k = 2)
sd[2] write (C, F) (k = 2)

Raid0 will then pick the largest service time generated from the array of SingleDisks and return it as the service time of the I/O request. Adding time spent queueing yields the overall I/O request response time.

RESULTS AND ANALYSIS

We now validate our simulator by comparing cdfs of generated response time with actual measurements from RAID 0 systems with 2, 3 and 4 disk drives. In all cases, we used an Infortrend A16f-G2430 RAID system containing four Seagate ST3500630NS disks. We set the stripe width on the array to 128 KB and disabled the write caches of both the disk drives and the RAID system. For each test, we read or wrote k blocks (1 ≤ k ≤ 20) for arrival rate λ = 0.01 requests/ms. The sample size for constructing a cdf is 5000 requests.

2-Disk Results

![Graphs showing 5-block read and 10-block read cdfs for 2-disk RAID 0 system with λ = 0.01 requests/ms.](image)

Figure 4: Selected response time cdfs on a 2-disk RAID 0 system for λ = 0.01 requests/ms.

Fig. 4 shows a selection of read and write results (presented in terms of the cdf of I/O request response time) for a number of request sizes on a 2-disk RAID 0 system with λ = 0.01 requests/ms. We observe close agreement between simulated and measured results for reads at all block sizes, while for writes we note that the simulation tends to slightly underestimate the measured results in all cases. This suggests that there is an additional overhead inherent in writes for which our model does not account. This discrepancy was also visible in the case of the single disk model, which suggests that the source is likely to be at the level of the disk rather than the RAID controller. Understanding the reason for this difference is a key area of our future work.

3-Disk Results

![Graphs showing 5-block and 10-block write cdfs for 3-disk RAID 0 system with λ = 0.01 requests/ms.](image)

Figure 5: Selected response time cdfs on a 3-disk RAID 0 system for λ = 0.01 requests/ms.

Fig. 5 shows a selection of read and write results for a number of request sizes on a 3-disk RAID 0 system with λ = 0.01 requests/ms. We observe close agreement between simulated and measured results for reads at all block sizes except k = 10, while for writes we note again that the simulation tends to slightly underestimate the measured results in all cases.

4-Disk Results

![Graphs showing 5-block and 10-block write cdfs for 4-disk RAID 0 system with λ = 0.01 requests/ms.](image)

Fig. 6 shows a selection of read and write results for a number of request sizes on a 4-disk RAID 0 system with λ = 0.01 requests/ms. We observe similar trends in the results as for the 3-disk case.
REFERENCES


CONCLUSION

In this paper we have presented a simulation model which predicts the cdf of I/O request response time in a RAID 0 system consisting of homogeneous zoned disk drives. We first constructed an M/G/1 simulation model of a single disk drive, which we validated against device measurement, and then used a split-merge queuing network to model the RAID system. We validated our resulting RAID 0 simulator against device measurements and demonstrated its accuracy.

There are a number of avenues for future work. Firstly, we need to account for the difference observed between our simulator and the measured results for write operations. Secondly, our simulator currently only models RAID 0 but there are several other commonly-deployed RAID configurations, particularly RAID 01 (mirrored stripes) and RAID 5 (distributed parity), which it could be extended to represent. Finally, our simulator is currently only capable of analysing request streams composed entirely of reads or entirely of writes. In the real world, streams of I/O requests will almost always be composed of a mixture of both, and we need to extend our simulator to be able to model this. Real I/O traffic can also consist of variably-sized requests and typically exhibits considerable burstiness, neither of which are supported by our simulator. We will therefore work to add support for these behaviours.
A New Generation of Nuclear Power Plant Simulation Applications

Mireya Rodríguez, José A. Cortés, Manuel Díaz, Daniel Garrido, Enrique Soler
University of Málaga
Department of Languages and Computing Science
CP.29071 Málaga, Spain
{jacortes, mdr, dgarrido, mireya.rodr, esc}@lecc.uma.es

ABSTRACT

Web Services is a new technology that uses standard transport and encoding protocols to exchange information. Its main feature is its interoperability, given that it allows communication among systems from any platform across intranets, extranets or the Internet, supporting at the same time a safe and reliable service. This paper presents our experience of using this technology in the context of Distributed Nuclear Power Plant Simulators and more specifically, in several subsystems called Instructor Console and Interactive Scenarios Manager. The former uses a modular mechanism based on dynamic plugings and the latter is a new application which, using Visual Studio Tools for Office, extends the use of an Office document to .NET applications.

INTRODUCTION

Some years ago, nuclear power plant simulators were based on old techniques and languages hindering scalability and reusability, mainly because of their dependence and coupling to the platform, the language and operative system used among other factors. The application of software engineering methodologies allowed the development of these simulators to be improved with concepts or technologies such as object-oriented programming, software components or, more recently, middleware. In previous work, (Díaz, M. and Garrido, D. 2004) we presented a new generation of nuclear power plant simulators with communications based on the Common Object Request Broker Architecture (CORBA).

CORBA is a communication middleware that allows the communication of objects developed in different programming languages and running on different hosts or operating systems in a transparent way. By using CORBA and software components, we defined a reusable and scalable communication architecture where new components could be added to existing simulators. We thought this architecture would be enough for the necessities of these simulators. However, we have recently developed several applications for these simulators with new requirements, generality, more flexibility, etc. which has motivated us to consider using new technologies which are more suitable for these necessities. More specifically, we considered the use of Web Services (WS) (W3C Working Group, 2007).

The main purpose of the simulators is to train future operators to carry out operations and maintenance, allowing different situations to be rehearsed, from the most usual situations such as temperature monitoring, valve operation, etc. to the most unusual emergency type situations. The objective of this paper is to select a communication technology suitable for our needs, taking into account different factors such as performance, interoperability and reusability.

At this stage/Here we present two new simulation applications: the Instructor Console (IC) and the Interactive Scenarios Manager (ISM), which can be used in any training centre to train future workers. This software has been developed by the company Tecnatom S.A. and the department of Languages and Computing Science at the University of Málaga. The work has been related to the development of simulation software with special emphasis on the use of new technologies.

SYSTEM ARCHITECTURE

The simulation projects underway in Tecnatom (Díaz et al. 2004) usually include two simulators that influence the hardware and software architectures. The first is called the Interactive Graphic Simulator (IGS), which through graphic applications (see Figure 1) allows future operators to be trained.

The second simulator called the Full Scope Simulator (FSS), is an exact replica of the control room of the power plant (see Figure 2). These simulators share parts of the hardware architecture and a large part of the software components, especially the components related to communication.

Fig. 1. Example of IGS application
The main hardware elements of FSS and IGS are:

- **Simulation Computers:** These computers are responsible for the simulation process executing the simulation models and providing data to the rest of the software and hardware components. They are the main elements of the simulators.

- **Subsystems:** Depending on the nuclear power plant, there will be several subsystems that have to be simulated.

- **Student workstations:** The IGS simulators additionally include the hardware needed for the student posts of the simulator.

Concerning the software architecture, simulators comprise a collection of distributed applications that can be executed on different nodes of a network, interacting with each other through the high-level communication mechanisms of CORBA. As a main goal, all applications and libraries have been developed in such a way that they can be reused in other simulators. Due to the implementation of a suitable, CORBA-based communication infrastructure, new applications can be added to the simulator without modifications in the software architecture. Some of the most important high-level applications together with their interactions can be seen in Figure 3.

The following is a brief description of their functionality.

- **Simulator kernel.** To compute the simulation of the power plant, the simulator kernel executes simulation models and calculates a great deal of simulation data. Through an attached CORBA-based communication layer, the kernel offers a set of services such as periodic transfer of variables, actions carried out on the simulator, etc. to the other applications.

- **Client applications.** They are a wide group of online applications that communicate with the simulator kernel for many different purposes, such as debugging the simulation process, allowing representation and modification of simulation variables, changing simulation aspects such as cycle time, recording or restoring the simulation state in real-time, etc.

In the rest of this paper, we will focus on the instructor console and the Scene Manager: architecture, functionality and how, by applying Web Services, previous versions together with performance tests can be improved.

**TECHNOLOGY SELECTION FOR THE COMMUNICATION**

When we started our study of the requirements of these applications we had to decide which technology to use taking several requirements into account. Some of the new requirements are the following:

- Reusability: different versions can be developed with a few variations. So the IC and the ISM can be used in different simulators.

- Scalability: it must be possible to add new functionality without changing the kernel of the applications.

- Heterogeneity: the IC must be used in a different execution environment such as standard PCs or Personal Digital Assistants (PDAs).

- Implementation in C# (required by Tecnatom S.A.) and the .NET platform.

- Intranet and internet versions.

- The IC and the ISM may require up to 4000 double digits per second from the simulator.

To achieve these goals we must evaluate the following technologies:

- **Sockets:** this was the technology used in previous versions of the IC. The main advantage was its efficiency. On the other hand, it lacks heterogeneity, is difficult to reuse, etc.
- **CORBA**: this is the technology used in the latest versions of the simulators. The integration of new components is easy, reusability can be obtained and heterogeneity is inherent in this approach.
- **Web Services**: it shares some of the main advantages of CORBA such as heterogeneity, scalability or reusability. In addition, it can be easily used in .NET with C#, and with new tools such as Visual Studio and VSTO. Finally it is important to take into account that the Web Services learning curve is smaller than that for CORBA.

Consequently, we decided to use WS in our training applications. The main objective here was to achieve all the use modes needed avoiding too much extra programming work and reusing as many components as possible. The WS technology is capable of supporting all the versatility required here as it is possible to develop just one version that can be used by different kinds of clients such as another web service, a windows form or a web page. The applications can be locally executed if installed first in the machine or accessed through a web browser from any device connected to a network.

### WEB SERVICE-BASED ARCHITECTURE

Nowadays the development of distributed applications running on different network connected machines is becoming more common. Typical technologies making this communication possible are DCOM (Microsoft Distributed Component Object Model) or CORBA which are reliable and scalable to some extent in an intranet environment but not over the Internet. The main reasons are that they have some problems with interoperability and their communication architectures (explicit invocation methods) lead to strongly coupled systems. In previous work (Diaz M. et al. 2007), an approach for simulator development based on RT-CORBA is presented.

Simulators are composed of several soft real-time applications. These simulate the behaviour of the Control Room of the Power Plant, where CORBA is the basis for the communication inside the simulator kernel and Web Services will be used in new applications such as the ones presented, where there are no real-time restrictions. In the following sections, the main contributions of using Web Services in the field of distributed simulators will be described.

Nonetheless, Web Service applications expose their functionality on the Internet, which makes these kinds of applications loosely coupled (the client need not know anything about the service implementation to which (s)he is gaining access, except the WSDL definition) and outstandingly interoperable.

The application of WS to the simulator allows us to benefit from all these advantages in the software development, together with all the interoperability and security mentioned. It could be argued that WS is not a sufficiently tested platform for critical systems. However, in these simulators we only use Web Services for communication in non-critical applications. In this sense, the loss of performance is not important compare to the point of view of real-time systems where the response time for an invocation could be higher. Of course, we are talking about response time of less than one second.

Once we decided to introduce WS to the whole system, a problem arose when trying to obtain data information from the simulator kernel since it is written in C and FORTRAN while WS uses C#. After searching for a way to intercommunicate applications executed at different times across different systems and networks that may be temporarily offline, the best solution found was the Microsoft Message Queue Server (MSMQ), figure 5. As
mentioned on its web site (http://msdn.microsoft.com/en-us/magazine/cc163920.aspx), MSMQ is a message infrastructure and a development tool for building distributed message applications for the Microsoft Windows Operating Systems.

Client applications call web services and the web server sends messages using MSMQ to the simulator. MSMQ uses message queues to ensure messages reach their destinations. MSMQ guarantees message delivery, reliability, safety and priority use in the messages. It also allows communication among applications written in both C and C# using .NET, as both languages have message send and receive primitives.

Using Microsoft Message Queuing. Due to the efficiency of communications and the fact that the queuing system allows asynchronous communication on the transmitter side (the simulator will not be blocked while waiting for the receptor - the web service - to read the message), it has been decided to use the Windows Queuing Messaging System (also known as MSMQ) instead of the other possible technologies such as sockets. Both C and C# languages contain send and receive primitives for MSMQ. To send information to a MSMQ system the MQSendMessage primitive is used, which sends a message to the queue previously specified. To receive data, the primitive to be used is Receive, which writes the first available message in the queue referenced by MessageQueue. This call is synchronous and blocks the current execution subprocess till a message is available. This is a minor disadvantage of the new architecture compared with the previous version. In the case of CORBA, the use of MSMQ is not required due to the language independence of CORBA. In this sense, the developed CORBA servers directly communicate with the simulator kernel developed in C++.

System Operation. Each second, (refresh frequency) the Simulator sends the simulation variable values to the message queue using the primitive MQSendMessage. Before sending these values, it checks if there are any unread previous message in the queuing system (variable values previously sent) and, where any exist, it deletes these values from the system. This behaviour is essential since each time variable values are updated, the previous ones are no longer needed and can be substituted by the new ones. Concurrently, web services, in response to the instructor console petition, request the current variable values, updated by the simulator using Receive from the queuing system. Later, the instructor console will use this information to show the simulator state. The instructor console will also send other data to the simulator such as a new value for the backtrack interval variable, as an instance.

At this point, we present two of our developed tools. They are designed to manage the simulation plant easily, controlled by an ordinary computer or a PDA, in such a way that interaction between simulation scenario–alum will be fully controlled by the instructor.

INSTRUCTOR CONSOLE

The Instructor Console (IC) is the client application that allows the instructor to supervise and control a simulation session. The IC will be capable of attending to the connected peripherals, communicating with the simulation programs and fulfilling functional requirements.

One of the main goals of this project is to develop a generic Instructor Console that can be easily integrated in any training simulator centre. To achieve this goal, an interface is used depending on the specific simulator it will be applied to. This interface will also be in charge of remarkable communicating the orders from the console to the simulator in the required format.

The global architecture this project is integrated into can be seen in figure 7. The system follows a client-server architecture. In the server part is the simulator with the attached simulation variables. The IC is located in the client side together with an intermediate console which will offer the IC an interface with the functions offered by the simulator. There is also a local copy of the most used variables to speed up the access to them.
The IC is composed of a main core to which several plugins, representing the different functionalities offered to the user, are added. This structure makes the software extensible easier and it isolates the integration of the functionalities from the rest of the code. The software is developed using Microsoft Visual Studio .NET C#.

The IC can be used according to the next using modes:
- Fixed local or network.
- Mobile local (PDA).
- Remote (Internet).
- Simultaneousness of uses (two different consoles in a same workstation). Note that if two consoles are attached to the same simulator, the state of the simulator is shared between them, but the variables to be communicated may be different.
- On-line use (with the simulation workload) and Off-line use (without the simulation workload by reading historical variables, actions and other data files, recorded in previously executed scenarios. [Aquí falta cerrar el parentesis, pero no sé dónde....]

**INTERACTIVE SCENE MANAGER**

This application complements the IC, because it is focused on the students who are being instructed. It is a new idea of communication, because the user’s interface is simply a Microsoft Word document. Training scenes are designed to achieve particular didactic objectives. Traditionally these training scenarios were included in “The Instructor Book”. The idea of the application is to translate each scenario of this book to a Word document. The innovation comes from the interactive nature of these documents with the simulation and with other user applications. In this sense, the objective of the tool is to ease the design, execution and evaluation of the instructor’s training sessions.

Therefore we have two profiles in this application. One is off-line with the simulator and is when the instructor can design the exercise. The other is on-line with the simulator, and is the part of the application which is in contact with the WS. Every action defined in edition mode can be executed by the students, interacting with the simulation session. Some controls send information to the simulation and others receive it, using a WS as communication interface, so that students can know the state of the simulation just by looking at the document.

The technology used in this tool is Visual Studio Tools for Office, which defines a package of .NET libraries for Office. It allows us to create Office projects with most of the Windows applications elements, and programming within the framework of the Object Oriented paradigm. Specifically, ISM is an Application-Level add-in developed by Microsoft Word 2003, and it is extensible to Custom Task Panes. This means having an application-specific functionality in a fully customized task-pane, where users can dock into different areas of the screen.

**Fig 8. the ISM Architecture**

When an instructor wants to define a specific simulation scene s/he can use an ordinary Word document in his/her machine and customize it. The word document can be accessed through a Windows application, which is in charge of linking the assembly and the deployment manifest, who are been generated by the Microsoft Office application deployed on VS and VSTO (Figure 8). When the link has been done, the document contains all the logic and the information for subsequent sessions. In edition mode, the instructor defines a specific scene with some training objectives, and for that uses the Word task pane, where there are controls for some actions allowed in the simulator (Figure 9). The students open the document with a double click, and it is then ready to be executed, on-line with the simulator. During the session they are going to send and gather specific information to the scene. (Figure 10)

The ISM actually covers the following aspects of the training exercise:
- Loading an initial condition to start a specific simulation scenario.
- Saving an initial condition, to create a scenario.
- Introducing anomalous situations in the simulation to train the student to manage them (called Malfunctions).
- Introducing a set of parameters in the simulator.
- Visualizing the state of simulation variables in the document.
- Visualizing tendency graphics to simulation variables.
- Allowing to stop/start the simulation.
- Allowing the instructor (only) to set the group of simulation variables which are loaded, from an XML file, in order to setup the document in many simulation environments.
CONCLUSIONS

Many companies, while trying to adapt to the rapidly changing technology, study how to take advantage of the
new research and technologies available.

In this paper, the adoption of new technologies such as
Web Services and VSTO has been presented in a project
for using the Instructor Console and the Interactive Scene
Manager of Nuclear Power Plant Simulators. The main
contribution is the utilization of Web Services in
Distributed Simulators. In this way, the development of the
communication infrastructure is easier than with traditional
techniques such as CORBA or sockets. Traditional
communications would be substituted by sockets with a
newer, considerably extended technology, more reliable
and efficient and obtaining significant advantages, overall
in scalability and application maintenance.

In this paper, we present the motivation for using WS,
which is basically the availability of Internet and
interoperability with the .NET platform and C#. If our
choice had been CORBA or Sockets, we would possibly
not be able to expand the training applications towards
Microsoft Office customizations, and it would be difficult
to achieve adaptability of the Console both for Internet and
PDA.

In conclusion, if performance is not mandatory, Web
Services may be the best solution. On the other hand, in
critical systems where optimal performance or a
predictable behaviour is needed, we argue that CORBA is
the best solution. Depending on the particular scenario, we
will choose the technology to be applied according to the
specific subsystem interacting with the simulator. In this
way and by combining both technologies we are
implementing new, valued-added services.

REFERENCES

Buyya, R.: Grid Computing Info Centre. Online documents at

Dahmann, J.S.; Fujimoto, R. and Weatherly, R.M: The
Department of Defense High Level Architecture, Winter

Diaz, M., Garrido, D., Troya, J.M: Development of Distributed
Real-Time Simulators Based on CORBA. Simulation
Modelling Practice and Theory. Vol. 15, num. 6, 716-733
(2007)

Diaz, M., Garrido, D., Soler, E: A Distributed Simulation Tool
on PDA. Proceedings 14th IASTED International Conference
on Applied Simulation and Modelling, pp. 376-381, (2005)

Diaz, M. and Garrido, D: A Simulation Environment for Nuclear
Power Plants. 8th IEEE International Symposium on
Distributed Simulation and Real Time Applications, pp. 98-

Henning, M., Vinoski, S.: Advanced CORBA Programming with
C++, Addison-Wesley Longman (1999)

Lankes, S., Jabs, A., Bemmerl, T.: Integration of a CAN-based
Connection-oriented Communication Model into Real-Time
CORBA: In IEEE International Parallel and Distributed
Processing Symposium (IPDPS 2003), Nice, France, April
(2003)

Levine, D.L., Mungee S., Schmidt, D.C.: The Design of the TAO
Real-Time Object Request Broker. Computer

Mark Pullen, J., Brunton R., Brutzman, D.P., Drake, D., Hieb,
M., Morse, K.L., Tolk, A: Using Web services to integrate
heterogeneous simulations in a grid environment. Journal of
Future Generation Comp. Syst., vol. 21, num. 1, pp. 97-106,
(2005)

Message Queue Server, Microsoft Technet. Online document at
http://msdn2.microsoft.com/en-us/library/ms711472.aspx,
(2007)

Object Management Group: The Common Object Request
Broker: Architecture and Specification, 2.6 ed., December
(2001)

Powell, M: Using ASP.NET Session State in a Web Service”.
Microsoft Corporation Online document at
(2002)

Schmidt, D.C., Kuhns, F: An overview of the Real-time CORBA
Specification. IEEE Computer special issue on Object-
Oriented Real-time Distributed Computing, June (2000)

Shodja, P: Web Services and the Microsoft Platform. Microsoft
Corporation. Online document at
(2006)

W3C Working Group, Web Services Architecture. Online
SIMULATION
AND
AI
INDUSTRIAL APPLICATIONS
DIAGNOSIS OF EXTRACTION SCREEN PLUGGING IN CONTINUOUS COOKING APPLICATION

Timo Ahvenlampi and Miika Uusitalo
Systems Engineering Lab., University of Oulu, Finland
email: timo.ahvenlampi@oulu.fi

KEYWORDS
Continuous cooking, diagnosis, extraction screen, Industrial processes

ABSTRACT

In this study, the operation and plugging in the extraction screens in the continuous Downflow Lo-Solids\textsuperscript{TM} type cooking application is investigated. The plugging occurs in the surface of the screen in the extraction flow and could prevent its normal operation. That can also lead to the shutdowns of the digester. The aim of the research is to diagnose different operation states in the continuous cooking application using several methods. Industrial data is used in the evaluation.

1 INTRODUCTION

Industrial applications, like continuous digester process, are very difficult to model, diagnose and control due to the highly non-linear nature of process. The continuous digester is complicated system with long residence times. The measuring of temperature and alkali content in the cooking plant is difficult and often also impossible, due to the high pulp capacity demands and highly alkaline conditions inside the digester. Thus the measurements are located at the input and output flows and on the outside wall of the digester. The production rates in the continuous cooking processes have increased continuously (Gullichsen 2000). That set demands for process control and diagnosis and can cause faults and disturbances which reduce the quality of the pulp.

The fault diagnosis of the chemical processes is an important factor in the quality control of the processes. Diagnosis of the chemical processes are studied in many papers, see e.g. (Dash et al. 2003), (Iserrmann 2005) and (Uraikul et al. 2007). Model-based methods are studied in paper (Iserrmann 2005). Fuzzy logic was applied for the fault diagnosis in (Dash et al. 2003). Survey of several methods was made in paper (Uraikul et al. 2007). In large scale systems, every component has its own task and the overall system works satisfactorily only if all components provide the service they are designed for. A fault in a single component usually changes the performance of the overall system. Therefore, faults have to be found as quickly as possible and decisions that stop the propagation of their effects have to be made (Blanke et al. (2003)). Active Fault Tolerant Control (AFTC) detects and isolates possible faults in the system and also reconfigures the control law (Mahmoud et al. (2003)).

The aim of this study has been to improve the diagnosis of the extraction screen plugging in the continuous cooking digester. In the studied plant, the plugging happens generally in the beginning of the hardwood cooking. In this study, three approaches are applied for the diagnosis of the normal and abnormal operation states. The squared extraction flow versus pressure difference over the extraction screen is one approach for the diagnosis. This has been used in the screening rooms, see e.g. R. Gooding (1993) and (Ahvenlampi et al. 2006) but there are not presented papers for the continuous cooking process. In the second approach, the current status of the process is shown using diagnosis plot. In the diagnosis plot, compaction (chip bed pressure) in the digester is used. In the third approach, fuzzy clustering is used for the diagnosis. In this study, the plugging is divided into several states and these states are presented in different colors. This signal is visualized using color codes. The traffic light colors are used. If the system is in a good process state, the signal is green. A slight deviation from the normal process state is indicated using a yellow color, and very significant changes are colored with red.

In following section, studied process is introduced. Used methods are revised in next section. Then, the results are shown and compared and finally discussion and conclusion are presented.

2 PROCESS

Studied application is a Downflow Lo-Solids\textsuperscript{TM} (Marccia 1996) cooking process (Fig. 1). The chips are impregnated in the impregnation vessel (I1-I2) and in the first zone (D1) of the digester. Between upper extraction and cooking circulation there is a counter-current zone (D2). In this zone, black liquor is displaced with cooking circulation liquor which temperature and alkali concentration are high. The lignin is mainly removed in the comparatively long co-current cooking zone (D3). The
black liquor is taken out from the digester through the lower extraction screen, which is situated below the co-current cooking zone. The plugging takes place in this screen and it is diagnosed in the study. At the bottom of the digester there is a short washing zone. Softwood chips mainly consist of pine chips with a small amount of spruce chips. Hardwood chips consist mainly of birch chips.

![Diagram of continuous digester](image)

Figure 1: Continuous digester.

**3 PLUGGING OF THE SCREEN**

The plugging of the screen is a problem in the continuous cooking application. The plugging happens generally in the lower extraction screens due to the high compaction degree (low Kappa number and high chip bed pressure), which hampers the black liquor flow through the screen. The main reasons for the plugging are high packing degree after the cooking zone, increased liquor flow and increased velocity difference between chips and liquor (Lee and Bennington 2004).

The prediction or elimination of the plugging can prevent shutdowns and increase production. In this study, the plugging of the extraction screen is studied during the operation of hardwood and softwood species.

Due to the plugging, the control limit is usually set to some fixed pressure differential value in the industrial screening plants (see Fig. 2). The same approach is also applied in the continuous cooking plants. However, this is not valid in all situations. One method to observe the plugging is to present the squared accept flow versus pressure differential (PDI) of the feed and accept flows in the screening room, as can be seen from Fig. 2. In the continuous cooking application, same kind of method can be used, see Fig. 3. If the screen is plugging, operation point is not in the plugging line but over it. (R. Gooding 1993) This can occur below the fixed differential pressure.

**4 DIAGNOSIS METHODS**

In the first approach, the plugging lines are used in the diagnosis of the plugging of the upper and lower extraction screens in the continuous cooking application. The plugging is divided into several operation states. The lines in Fig. 3 show the states of the lower extraction screen plugging. If the data point is below the yellow (solid) line the state is good and the plotting is done using green color. If the measurement data is between yellow and red (dashed) line the state is little worse and there is some plugging happening in the extraction screen. If the data point is between red and black (dotted) line the state is worse and the there is a lot of plugging in the screen. Finally, if the data is over the black line the state of the screen is bad and the digester could be shutdown for a while.

In the second approach, the current compaction state is presented in the different zones of the digester. The basic idea of the approach is shown in Fig. 4. The idea is to diagnose over and under compaction in different zones of the digester. The lower and upper limits for the
compaction (chip pressure) are shown with solid lines and the current status of the digester is plotted using dashed line and it can be used to diagnose problematic operation states. The compaction changes in the top of the digester affects to the compaction in the bottom of the digester. Information of the compaction change can be used in the diagnosis of possible plugging problems in the lower extraction screen.

In the third approach, the fuzzy clustering (Gustafson-Kessel algorithm (Gustafson and Kessel 1979)) is applied for the diagnosis of the studied process. The clustering data (55 variables) is a collection of the measurements from the automation system and modelling results. The clustering is applied especially for the faulty operation periods.

Gustafson-Kessel (GK) algorithm (Gustafson and Kessel 1979) is the mostly used extension of the Fuzzy C-means (FCM) in identification (Babuska 1998). In this method, norm can be different with every cluster and method has the advantage of looking for variable size ellipsoids:

\[
D^2_{ik,B_1} = (z_k - c_i)^T B_1 (z_k - c_i)
\]  

(1)

In this way, quasi-linear behaviors of the existing operation states are detected quite correctly.

5 RESULTS

In this section, the diagnosis and simulation results are presented.

In Fig. 3, the squared extraction flow versus the pressure difference over the extraction screen is shown. This is a good validation period, which is a typical situation in the studied plant. Sometimes there is still some problems with the screens and this kind of validation period is presented in Fig. 5.

In the good validation period (Fig. 3), pressure difference over the lower extraction screen increases in a linear fashion with squared extraction flow. When extraction screen plugging occurs (Fig. 5), measured pressure difference points rise up from the normal operation point and overall graph loses its linearity.

One reason for the plugging of the lower extraction screen was found to be the inner flows in the digester. In Fig. 6, is presented the ratio of the upper and lower extraction flows in good validation period and in Fig. 7 in faulty operation period.

Other reason for the plugging is the density differences of the penetrated chips and liquor in the digester. In Figs. 8 and 9, are presented density differences at the top of the digester in the good and faulty operation period.

Figure 3: The squared extraction flow versus the pressure difference over the extraction screen in good validation period.

Figure 4: The compaction in the different digester zones.

respectively. The density difference and chip and liquor level difference in the top of the digester affects to the chip bed pressure (chip compaction).

In Figs. 10 and 11, are presented the squared upper extraction flows versus pressure differences over the upper extraction screen.

In Fig. 12, is presented the diagnosis plot for the softwood case. In this case, the compaction in the cooking zone D3 is over the limits and it indicates that there might be problems in the lower extraction screen. The diagnosis plot for the hardwood case is shown in Fig. 13.
Clustering results for the faulty operation period are presented if Figs. 14 and 15. The trend in the Fig. 15 is shown using the clustering results in Fig. 14.

6 DISCUSSION

The plugging of the extraction screen has an major effect to the operation of the cooking process. The elimination of the plugging problem can increase the production in the plants. In this study, the plugging of the lower extraction screen is studied using diagnosis. In Fig. 3, is presented good validation period from the plant.

Extraction screen plugging derives from chip column’s high packing degree on the screens. While chip column’s packing degree increases, open surface of the screen decreases resulting increasing flow resistance over the screen. Increased flow resistance can be seen as a rise in differential pressure over the extraction screen. High packing degree results from high chip bed pressure and low kappa number. At the level of the lower extraction screen, chip bed’s height is nearly 40 meters, thus chip bed pressure is high, and kappa number is dropped to the target blowline kappa number.

Chip bed pressure consists from density differences between the chips and free liquor, friction between chip column and digester wall and the liquid flow resistance resulted from velocity differences between the chips and free liquor. In the top of the digester, also the difference
between chip and liquid levels has an affect for the chip bed pressure. The chip bed pressure in the bottom of the digester is cumulated from the upper part of the digester. Thus, the chip bed pressure changes in the top of the digester transfers via chips to the next point of contact. The chip pressure (contact forces) cannot be measured and it is average of the contact forces over the studied surface. (Härkönen 1987)

In this study, possibilities of decreasing chip pressure to reduce plugging of lower extraction screens, have been investigated. In Downflow Lo-Solids$^{TM}$ cooking, digester’s internal liquor flows can be modified more freely than in conventional digester. Lowering the density difference between the chips and free liquor would be the most effective way to decrease chip pressure, but in a manner of maintaining cooking chemistry this can be very difficult. Consequently the most reasonable way to modify chip pressure in Downflow Lo-Solids$^{TM}$ cooking is to control digester’s inner liquor flows. In Figs. 6 and 7 are presented upper and lower extraction flow ratios in good and faulty operation periods, respectively.

Lets introduce a theory of lowering the chip bed pressure by controlling digester’s inner flows. By increasing inner liquor flows in a counter-current zones of the digester (zones D2 and D4) and decreasing liquor flows in co-current zones (zones D1 and D3), the chip pres-
Figure 13: The compaction in the different digester zones in hardwood pulping.

Figure 14: The squared extraction flow versus the pressure difference over the extraction screen in faulty validation period (clustered using Gustafson-Kessel algorithm).

Figure 15: Trend of the pressure difference over the extraction screen in faulty validation period (clustered using Gustafson-Kessel algorithm).

extraction flow, zone’s D2 counter-current of free liquor has been about 40% higher on a hardwood cooking than in validation period where the screen plugging occurs. Also the co-current flow of the zone D3 is 50% lower in the good period than in the faulty period.

In Figs. 8 and 9 are presented the density differences in the good and faulty validation period 2. The density difference change (10 kg/m³) in the upper part of the digester changes the chip pressure about 500 Pa. In figures 8 and 9, the colors indicate the plugging line state. It can be seen that the bigger density difference at the top of the digester can increase plugging of the lower extraction screen. The density differences are mainly affected and changed by the alkali and temperature control actions.

In this study, also the plugging in the upper extraction screen is considered. As can be seen from Figs. 10 (good validation period) and 11 (faulty validation period), the plugging is not problem in the upper extraction screens. The reason is that the chips are not delignified a lot (Kappa number is high) in that part of the cooking process and therefore the chips are not as compacted as in the lower extraction screen.

The proposed plugging line diagnosis system will be implemented into the plant’s automation system, so the operators get more information for the diagnosis of the plugging of the lower extraction screen.

In the other approach, the plugging of the lower extraction screen is diagnosed using chip bed pressure which is presented in the different digester zones, see Fig. 12 for the softwood case and Fig. 13 for the hardwood case.
In Fig. 12, there is over compaction (too high chip pressure) after the cooking zone (D3). This is illustrated also using red color in the plot. In the other zones, the compaction is between the lower and upper limits (this is shown using green color).

In the hardwood pulping example (Fig. 13), the compaction after the zone D2 is over the upper limit (for instance too low Kappa number in the zone D2). After the residence time in the zone D3 the over compaction will be seen in the zone D3, if the process operation is not changed during that time. The proposed diagnosis plot system can be implemented into the plant’s automation system.

The clustering of the data indicated also problematic operation points in the lower extraction screen plugging. In Fig. 14, is presented example of the faulty validation period using Gustafson-Kessel algorithm. Seven clusters were used. The pressure difference over the extraction screen with clusters presented in Fig. 14 is shown in the Fig. 15. As seen from the Fig. 15, the pressure difference in the extraction screen is moving from the normal operation (time period 0-75) to the faulty operation (75-220) and finally the screen has plugged at the time 220.

7 CONCLUSIONS

The compaction and especially plugging of the lower extraction screen in the continuous cooking application is studied. Three approaches are presented for the diagnosis of the Downflow Lo-Solids™ type digester. In the first approach, the plugging of the screen is divided into several fault diagnosis states. These states are illustrated using color codes. In the other approach, the chip pressure in the different zones of the digester is illustrated. This can be used to diagnose the lower extraction plugging, because the compaction cumulates from the top of the digester to the lower part of the digester (to the lower extraction screen). In the third approach, clustering is applied for the diagnosis. Good diagnosis results have been achieved.

ACKNOWLEDGEMENTS

This research study was funded by The Academy of Finland. The data is from Oy Metsä-Botnia Ab. The authors would like to thank the partners for the special knowledge and process data provided.

REFERENCES


INTERACTIVE SIMULATION SYSTEM FOR TESTING WASHING MACHINE DESIGNS

Carla Martin-Villalba  
Alfonso Urquia  
Sebastian Dormido  
Departamento de Informática y Automática, UNED  
Juan del Rosal 16, 28040 Madrid, Spain  
{carla,aurquia,sdormido}@dia.uned.es

Félix Martínez  
Area de Ingeniería Mecánica, IKERLAN  
Pº J. M. Arizmendiarríe 2, 20500 Gipuzkoa, Spain  
Felix.Martinez@ikerlan.es

KEYWORDS
Interactive simulation, object-oriented modeling, hybrid-models.

ABSTRACT
The design, implementation, validation and use of an Interactive Simulation System (ISS) for testing designs of drum-type washing machines are discussed in this manuscript. This ISS has been successfully applied in the design process of an industrial washing machine (40 Kg load capacity) manufactured by Fagor Industrial. It has been developed using two free Modelica libraries - MultiBody and VirtualLabBuilder - and the Dymola modelling environment. The MultiBody library has been used to compose the washing machine dynamic model, which has been validated using experimental data for transient and steady-state analyses. The interactive visual interface has been implemented using the VirtualLabBuilder library. This ISS has demonstrated to be a valuable design and analysis tool, allowing users to: (1) get insight into the system behavior; (2) tune the model parameters in order to improve the system dynamic behavior; (3) simulate special events, such as a component breakage; and (4) configure the properties of the suspension system.

INTRODUCTION
Interactive Simulation Systems (ISS) allow users, through visualization aids such as graphics and animations, to explore the simulation results, and to modify the model variables and parameters during the simulation run (Zhao 2002). An ISS is typically composed of two interrelated parts: the model and the view. The model is the simulation model describing the relevant behaviour of the system under study. The view is the user-to-model interface. It is intended to provide a visual representation of the model dynamic behaviour and to facilitate the user's interactive actions on the model during the simulation run.

The model and the view of the ISS discussed in this manuscript have been programmed using Modelica (Modelica 2008). The ISS model has been composed using the MultiBody Modelica library (Otter et al. 2003), and it has been adapted for interactive simulation following the methodology proposed in (Martin-Villalba 2007, Martin-Villalba et al. 2008). MultiBody is a free Modelica library, providing 3-D mechanical components to model mechanical systems.

The ISS view has been composed using VirtualLabBuilder (Martin-Villalba 2007, Martin-Villalba et al. 2008, VirtualLabBuilder 2008). This free Modelica library allows the ISS developer to:
- Compose the ISS view by instantiating and connecting the predefined components provided in VirtualLabBuilder, such as containers, animated geometric shapes, basic elements and interactive controls. The Dymola (Dynasim 2006) modelling environment allows developers to graphically compose the model. The connections among these elements determine the layout of the ISS view.
- Link the visual properties of the above mentioned graphic components with variables of the ISS model.
- Link HTML pages (for instance, containing documentation about the ISS design and use) to the ISS view.

The Modelica code describing the complete ISS is translated into an executable application using Dymola (Dynasim 2006). The ISS is launched by executing this stand-alone application, which only requires the Java Development Kit for running. By executing this application, the Java code of the ISS view is automatically generated, compiled and run. Then, the communication procedure between the ISS view and model is automatically established, and the interactive simulation starts. All the details are discussed in (Martin-Villalba 2007, Martin-Villalba et al. 2008).

The ISS presented in this article is intended to test a wide range of drum-type washing machine designs. Its aim is twofold. Firstly, to configure the properties of the washing machine suspension system (i.e., the location of the spring and damper extremities, and the value of the spring and damper constants). Secondly, to analyze the influence of some design parameters, such as counterweight mass locations and physical properties (i.e., mass distribution), in order to reduce forces and vibrations transmitted to the frame. These analyses allow estimating the minimum distance between the suspended drum and the frame, which is required to avoid the suspended drum to collide with the fixed elements, such as the soap dispenser.
The ISS has also proved to be useful for: (1) defining “rules of thumb” for designers; (2) tuning control parameters; and (3) evaluating risky situations, such as damper and spring breakage.

**MODELLING AND VALIDATION**

Drum-type washing machines, widely used in Europe, are composed of an inner drum that rotates inside an outer drum, with a horizontal axis, making the clothes tumble upward and downward during the washing cycle (see Fig. 1a). The Multibody Modelica library (Otter et al. 2003) has been used for developing the dynamic model of the washing machine.

The model diagram is shown in Fig. 1b. All the bodies, except the springs, have been considered as rigid bodies. The suspended drum is composed of seven bodies: outer and inner drums, and one centered, two uncentered and two counterweight masses. These masses are attached to the inner drum. A rotational degree of freedom is allowed between the inner and the outer drums, where the speed set point is applied. The suspended drum has 6 degrees of freedom. Its dynamics is governed by the forces generated by the two uncentered masses, gravity, and the forces exerted by N pairs of springs and dampers.

The springs and dampers are modelled as ideal elements (i.e., their forces proportional to displacement or opposite to relative speed, respectively). Their forces are applied in the directions defined by their extremities. Additionally, an external, massless, frame is considered. The springs and dampers are attached to this frame. This model allows the computation of the floor reaction forces.

The model has been validated for transient (i.e., spinning up speed) and steady-state analyses. In both cases, the simulation results are in good agreement with the measured experimental data. For instance, the simulation results and the experimental data for planar movements at 1100 rpm spinning speed are shown in Fig. 1c. The experimental data show better cycle-to-cycle repetitiveness than the simulation results because the signals, measured with piezoelectric accelerometers, were filtered at 1 Hz to remove the offset value.

**ISS DEVELOPMENT**

The washing machine model has been adapted for interactive simulation following the methodology proposed in (Martin-Villalba 2007, Martin-Villalba et al. 2008). The Modelica modelling environments, such as 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. The connections among these elements determine the layout of the ISS view.
The following two types of interactive parameters have been considered in the model:

1. Parameters that configure the properties of the washing machine suspension system (i.e., the location of the spring and damper extremities, and their coefficients).

2. Parameters that define a type of washing machine. These parameters are the following: number of pairs (N) of springs and dampers, physical properties -mass, centre of gravity and inertia- of the outer and the inner drums, and of the counterweight and uncentered masses.

The value of the first type of parameters can be changed by the ISS user at any time during the simulation run. On the contrary, the value of the second type of parameters can be set by the ISS user only before the simulation starts. The ISS can be used to analyze the behaviour of a wide range of washing machines (from small home appliances to big capacity industrial washing machines). This can be performed in a simple way, by modifying the value of these parameters.

The Modelica description of the ISS view has been developed in a modular way, by instantiating and connecting the required graphic components of the VirtualLabBuilder library. The ISS view contains one main window and seven dialog windows, each one described by a Modelica class. For instance, the Modelica class describing the main window of the ISS view is shown in Fig. 2a. It is explained below the relationship between this Modelica description and the Java window that it automatically generates (shown in Fig. 2b).

The MF component is a container of the MainFrame class. It is a container at the higher hierarchical level, intended to host other containers. The MF layout policy is set to BorderLayout, in order to select the placement of each hosted container out of five possible placements: north, south, center, east and west. MF has two container components hosted inside it (see Fig. 2a): pCenter and pSouth, both of the Panel class.

- pCenter is placed in the center of MF. It hosts two containers (see Fig. 2a) - DP1 and DP2 - of DrawingPanel class. These two containers host the frontal and the lateral animated diagrams of the washing machine, respectively. These diagrams are composed using some VirtualLabBuilder classes defining graphic components.

- pSouth container hosts other containers, that host components of VirtualLabBuilder classes implementing interactive controls: a combo-box, a set of buttons, a numeric box and a set of check-boxes.

When the ISS is launched, the view is displayed but the simulation does not start. The user can specify the initial value of the model parameters in a text file, which can be imported from the ISS before starting the simulation. This feature facilitates the definition of the initial experimental conditions. Also, the model parameters can be saved to a text file at any time during the simulation run. The ISS contains a “Save parameters” button that, when pressed, generates a text file including the name and the actual value of the model parameters.
Figure 3: Washing machine ISS: a) Dialog window showing the time evolution of the spring lengths; and b) Dialog window with sliders for changing the position of the damper extremities.

The generated text file can be imported from the ISS, in order to set the initial experimental conditions for a subsequent run.

The ISS contains a “play/pause” button that allows starting the simulation once the initial value of the model parameters has been set. In addition, this button allows the user to pause or resume the simulation at any time. The numeric box and the check-boxes allow the user to set the spinning speed value, and to show and hide the dialog windows, respectively (see Fig. 2b).

There are two types of dialog windows: (1) those containing plots of model variables; and (2) those containing sliders for changing the value of some model variables. An example of each window type is shown in Figs 3a and 3b.

There are windows containing plots showing the time evolution of the following variables: spring lengths, damper lengths, position of some points placed in the inner drum, position of the uncentered masses. The plot time-span is a parameter that can be set by the user. The time evolution of the spring lengths is shown in Fig. 3a. The user is allowed to show and hide the trails by pressing the corresponding check-box. This plot is automatically scaled in function of the maximum and minimum values of the displayed trails.

The “Dampers data” (see Fig. 3b) and “Spring data” windows allows the user to modify during the simulation run the location and constant value of the dampers and the springs respectively. The user is allowed to set the location of the springs/dampers by specifying their height and the location of their two extremities.

RESULTS AND DISCUSSION

The ISS has been successfully used to define the geometrical configuration of washing machines with different load capacities. A good washing machine design tries to minimize the amplitude of the spring and damper length changes, and the displacement amplitude of external points of the drum at different spinning speeds. The time evolution of these variables is shown in plots included in the ISS view (an example is shown in Fig. 3a). In addition, the ISS view contains animated diagrams of the lateral and frontal view of the washing machine. This visual information can be used to tune the model parameters, in order to define the geometrical configuration of the washing machine. This parameter tuning can be performed by using the ISS controls (some are shown in Fig. 3b). Finally, once the optimum values of the design parameters have been found through this interactive tuning process, they can be saved to a file by pressing the “Save parameters” button.
The use of this ISS has facilitated the early verification of the washing machine design concept. It allows performing the typical design analyses quickly and accurately. As a result, the ISS has contributed to shorten the design cycle time and to increase the designer productivity. In particular, the geometrical configuration of an industrial washing machine (40 Kg load capacity) manufactured by Fagor Industrial has been analyzed and its design has been optimized in about four working hours.

CONCLUSIONS

The design, implementation, validation and use of an Interactive Simulation System (ISS) for testing a wide range of drum-type washing machine designs have been discussed. It has been composed using two free Modelica libraries: MultiBody and VirtualLabBuilder. Dymola has been used to translate the Modelica description of the ISS into an Executable application, which only requires the Java Development Kit for running.

This ISS has been successfully applied in the design process of an industrial washing machine (40 Kg load capacity), manufactured by Fagor Industrial. The ISS use has demonstrated to facilitate: (1) faster parameter tuning; (2) the analysis of special events (e.g., spring or damper breakage); and (3) the evaluation of different suspension system configurations. In consequence, the ISS constitutes a very valuable tool for designers, that can online explore “what happens if” with no need to run again the simulation, enhancing their knowledge of the system behaviour.

ACKNOWLEDGEMENTS

This work has been supported by the IKERLAN - UNED contract “Laboratorio virtual de lavadora industrial en Java-Dymola” (November 2007), by the Spanish CICYT, under DPI2007-61068 grant, and by the IV PRICIT (Plan Regional de Ciencia y Tecnologia de la Comunidad de Madrid, 2005-2008), under S-0505/DPI/0391 grant.

REFERENCES


VirtualLabBuilder Modelica library web-site: http://www.euclides.dia.uned.es/VirtualLabBuilder

BEHAVIOURAL AI
Simulating and Monitoring Ambient Assisted Living

Davide Carneiro\textsuperscript{1}, Ricardo Costa\textsuperscript{2}, Paulo Novais\textsuperscript{3}, João Neves\textsuperscript{3}, José Machado\textsuperscript{1} and José Neves\textsuperscript{1}

\textsuperscript{1}DI-CCTC, Universidade do Minho, Braga, Portugal  
davide.carneiro@gmail.com, pjon, jmnc, jneves@di.uminho.pt  
\textsuperscript{2}College of Management and Technology - Polytechnic of Porto, Felgueiras, Portugal  
rfc@estgf.ippt.pt  
\textsuperscript{3}Centro Hospitalar de Vila Nova de Gaia e Espinho, EPE, Portugal  
j.neves@hotmail.com

KEYWORDS

e-Health, Ambient Intelligence, Remote Monitoring and Simulation.

ABSTRACT

Researchers are giving special attention to innovative healthcare projects in order to reduce medical service costs and to deal with the population ageing. The Virtual e-Care was born taking those goals in mind. An intelligent and proactive system has been prototyped, supporting group decision making techniques, idea generation, argumentation and the quantification of the quality of information. In this paper it is simulated a virtual Assisted Living Environment, based in a solid agent-based architecture. Special attention is given to the monitoring system.

INTRODUCTION

Healthcare systems around the world are suffering financial problems due to financial restrictions, social crisis and the population ageing. Healthcare budgets are short, quality is poor and citizens have no choice. Changes have to be made and it seems that technology and innovation are the only way to minimize those effects. Patients, especially the ones with chronic diseases, have to be empowered, thru the use of disease management programs, giving them control of their own situation and reducing medical costs. In this new reality, treatments are at home, thanks to the use of technological gadgets that allow real time monitoring and critical vital data evaluation. Alarms can be activated and recommendations will help people to overstep some basic problems. People will have access, in real time, to the collected information, giving them control about the actual health state of their relatives (IBM 2006).

Several Information technology (IT) healthcare projects to provide the above characteristics have arisen (Holmlid 2003), (Camarnha et al. 2003). Some projects are simple “panic buttons”, others domotics technologies with the intention of using intelligent sensors and, the most advanced ones, focus on the development of context-aware interfaces for the elderly or the development of “smart-homes” for aging-in-place. There are also generic frameworks for allowing the integration of the above projects, like the Telecare System. Our goal is to go a step ahead, the virtual-eCare.

AMBIENT INTELLIGENCE

Ambient Intelligence (AmI) is a new paradigm in IT, in which people are empowered through a digital environment that is aware of their presence and context, being this environment sensitive, adaptive and responsive to their actual needs, habits, gestures and emotions (Riva 2003). AmI can be defined as the merger of ubiquitous computing and social user interfaces. It is built on advanced networking technologies and formed by a broad range of mobile devices and other objects, adding adaptive user-system interaction methods, and based on new insights in the way people interact with computing devices (social user interfaces). As a result, digital environments can be created to improve quality of life of people by acting on their behalf. These context aware systems combine ubiquitous information, communication, and entertainment, with enhanced personalization, natural interaction and intelligence. The path to pursue, in order to achieve this goal, relies on a mix of different receptiveness from Artificial Intelligence, Psychology or Mathematical Logic (just to name a few), coupled with different computational paradigms and methodologies for problem solving, such as the conceptualization of figures like software agents, and its social counterpart in the form of Group Decision Support Systems (Costa et al. 2008).

INTER-ORGANIZATION COOPERATION

In Inter-Organization Cooperation, there are factors that tend to be surrounded in the local milieu, which according to (Dosi 1998), can be seen as the social embedded processes that allow organizations to obtain outside complementary knowledge and be innovative in the course of interaction among different actors, i.e., the local or regional milieu needs to include not only the substances related to the service structure or economies terms, but also social, cultural and institutional ones (Malmberg 1996). Thus, in the interaction of the different actors, the cooperation elements can be found in a kind of common languages, social relationships, norms, values and institutions, which in our work will be set in terms of an extension to the logic programming language, being their knowledge bases built as logical theories that found their foundations on this extension (Neves 1984). Conclusions are supported by deductive proofs, or by arguments that include conjectures and motivate new topics of inquiry. i.e., if deduction is fruitless the agent inference engine resorts to abduction, filling in missing pieces of logical arguments with plausible conjectures to obtain answers that are only partly supported by the available facts (to the inference engine).
SIMULATION

Simulation is the methodology of creating an alternative reality to represent a real object. When making simulation, generally one expects to predict how a given object or system behaves in the real world. So, it is possible to draw conclusions about the behaviour of the system under study and about its feasibility without building the system. It is important to select the most important characteristics or behaviours of the system to simulate, so that the results are as accurate as possible. These parameters to select may comprise not only the object being simulated but also the environment that surrounds the object. In the last years, this technique has grown a lot, mainly thanks to advances in computer system performances. Simulation is used to model complex systems that are too expensive, too dangerous or simply impossible to assemble in the real world. Some of the common uses of simulation are the modelling of natural systems (e.g. weather forecasts, storm evolution, earthquake damage), testing and optimizing new technologies, the construction of new or special buildings (e.g. the new skyscrapers, dams).

THE VIRTUAL-ECARE PROJECT

The Virtual-eCare Project (Costa et al. 2007) main objective is to present an intelligent multi-agent system able to monitor, interact and provide its customers with healthcare services of the utmost quality. This system is interconnected, not only to other healthcare institutions, but also with leisure centres, training facilities, shops and patient relatives, just to name a few.

The Virtual-eCare Architecture is distributed. Different modules are interconnected through a network (e.g. LAN, MAN, WAN), each one having a different role (Figure 1). A top-level description of the architecture machinery is given:
- SupportedUser – elderly with special healthcare needs, whose clinical data are sent to the CallCareCenter and redirected to the Group Decision Support System;
- CallCareCenter – the remaining ones being redirected to the CallServiceCenter; A special attention will be given this key module;
- Group Decision – it is in charge of all the decisions taken at the Virtual-eCare platform.

CallServiceCenter – Entity with all the necessary computational and qualified personal resources, capable of receiving and analyzing the complex data and taking the necessary actions;

CallCareCenter – Entity in charge of computational and qualified personal resources (i.e. healthcare and auxiliary professionals), capable of receiving and analyzing the clinical data, and taking the necessary actions;

Relatives – individuals that may have an active role in the supervising task of their love ones, being able to give precious complementary information about them and being able to intervene, in a complementary way, in specific crisis (e.g., loneliness).

In order to support Group Decision Support, the opinion of specialized staff (e.g., nurses, paediatrics, cardiologists) has to be collected. There is also the need to have a digital profile or history of the SupportedUser, allowing a better understanding of his/her special needs. In this profile, several relevant information arguments are imported from the patient Electronic Health Record, such as personal preferences (e.g. musical, gastronomic) and experiences. This solution will help healthcare providers to integrate, analyze, and manage complex and disparate clinical, research and administrative knowledge. It will provide tools and methodologies for creating an information-on-demand environment that can improve quality-of-living, safety, and quality of patient care.

VIRTUAL-ECARE INFRASTRUCTURE

We have designed a first proposal of a generic, configurable, flexible and scalable infrastructure as presented in Figure 2. It is expectable that on top, an extensive number of services will progressively arise. These services have been developed as Web Services, allowing the coexistence of heterogeneous software languages interacting through common messages. The fundamental components of the proposed infrastructure are:
- Secure Communications – in order to all the components interact, a secure communication infrastructure is mandatory;
- Management – responsible for configuring and monitoring the involved components;
- Resources – responsible for every component registration and managing the resource catalogue;
- Authentication – every component must authenticate in order to interact;
- Recommendation – responsible to make problem solving recommendations;
- Monitoring – responsible for interacting with all the sensors and reporting results to the GDSS;
VIRTUALE-CARE ARCHITECTURE

The Virtual-eCare architecture is composed by a series of different elements geographically separated (Figure 1). It is dynamic because elements can enter and leave at any time, logically or geographically, or the services they provide may vary. The main components of the architecture are the End User and its House, a Monitoring module, the Recommendation System, the Group Decision Support System, the Database Manager and a HL7 module (Hinchley 2007). Each element of the architecture may be very different in its functionalities and software languages. To achieve a distributed, modular, dynamic, extensible, flexible, scalable and compatible architecture, widely used technologies and standards have been selected, such as OSGi (Osgi Alliance 2003), R-OSGi, FIPA or Web Services (Monson-Haefel 2003).

To ensure communication and compatibility between different components, the Web Services protocol was used. Web Services ensure interoperability, information sharing and platform independence. Each component can offer a set of public or private Web Services that can be requested and available to the components. A component is at the same time a server and a client. The Recommendation System, for example, uses Web Services provided by both the House and the Database Manager and provides, as a service, the Recommendation system used by the Group Decision Support System.

The communication protocol is of major importance. As stated before, Web Services have been selected to implement synchronous communications between platform independent components. The information shared through Web Services and FIPA-ACL messages are represented in XML. This FIPA standard allows a description of the main content of messages without having to read content, using ontologies, specific languages or speech-acts. Messages are forwarded and sent to the final agents without checking contents.

Figure 3: Sample Communication Sequence Diagram

A particular way of structuring the actual content of the message in XML has been set. Content examples are the temperatures in a room or in an entire house (a list of rooms), the patient’s Electronic Health Record (EHR) or a recommendation coming from the Recommendation System. An example of a sequence of communication is presented in Figure 3. The bundle triggers the process, which is responsible for monitoring the vital signs of the Supported User. This bundle detects an irregular heartbeat and warns the House central OSGi, where the MAS is running using R-OSGi. The MAS requests information from the movement sensors in another OSGi and asks again about the cardiac rhythm to the bundle that started the process, to ensure that there was no reading error. Having gathered the information, the MAS decides that it cannot do anything to correct the situation and informs the Group Decision Support System, sending the anomalous values. This one contacts the Recommendation System, which reads the values of the sensors of the House and generates a recommendation, which is then issued back to the Group Decision Support System. After communicating with some more elements (like specialized doctors) and having in consideration the answer from the Recommendation System, two actions are taken: an ambulance is sent to the Home and Lights in the user’s room are turned on.

In Figure 3 Dashed arrows represent R-OSGi services being invoked, regular arrows stand for FIPA ACL messages being exchanged through Web Services and circles represent some major processing or communication with local bundles using OSGi. Due to lack of space, this picture is simplified.

In Figure 4 a simplified view of our architecture is looked. The arrows represent Web Services, which allow for the several components to exchange information. The arrows can be seen as “uses service from”, pointing from the client to the server. The House is more detailed, showing OSGi and R-OSGi sub-components interconnecting different kinds of elements.

Let us now detail the technologies used in the components by moving to a more close view of the architecture. At this level, two well known standards where used: OSGi and R-OSGi. OSGi is an initiative that intends to establish standards in Java programming, highly specific, catering for the sharing of Java classes, that may be achieved in terms of a service platform paradigm (Chen and Gong 2001). The use of this technology will let developers to build Java
applications on a modular basis. The resulting modules are called bundles, which are not only competent to provide services, but also to use services provided from other bundles. In OSGi, a bundle can be installed, started, stopped or un-installed at run-time and without system reboot, which makes OSGi-based technologies modular and dynamic.

[Diagram of Virtual-eCare Services]

R-OSGi is an extension to OSGi, which allows the access of services provided in remote OSGi implementations, in a completely transparent way, much like they were local services. OSGi and R-OSGi are used in our architecture to achieve two main objectives at the level of each component: to grant the compatibility and communication and to establish a logical organization inside the component. These issues come from the multitude of parts that each component can be made of.

Let us look, as an example, to the House component. The House is made of physical parts like 1-Wire sensor and X10 actuators, and logical ones like Multi-agent Systems (MAS). 1-Wire is used for measuring environmental values and X10 to control appliances and equipments. MAS is responsible for taking basic reactive actions, like control temperature or call for help. Moreover, the house may have a big number of rooms and floors, which can vary along the time. There is, firstly, the need to organize all these components logically. So, several OSGi implementations have been created. For each group of similar sensors in each room, a bundle has been created. This means that, for each room, there will be a bundle reading values from the temperature sensors, another one reading the values from the luminosity sensors and so on. These bundles provide, as a service, the mean value of the last series obtained from the respective sensors. As for the actuators, there is one bundle controlling each equipment or appliance, which is able to send X10 commands to the equipment it controls. The services these bundle provide are the X10 commands that can be issued.

The bundles of the same type in each floor run in the same OSGi implementation, i.e., in each floor there is an OSGi implementation for the temperature bundles, another one for the luminosity bundles, and so on. Likewise, there is an OSGi implementation for the appliances of each type on each floor, i.e., an OSGi for lights, another for air conditioning, etc. In addition, there can also be OSGi implementations inside the House that are just software, like the MAS that is responsible for taking the basic actions. The MAS is fully integrated with OSGi. Each OSGi implementation has at least one additional bundle: a R-OSGi bundle, which provides remote access to the implementation. This bundle acts as the bridge between the exterior and the sensors or actuators controlled by the implementation. Its services are the operations that can be performed on the components controlled by the implementation. In the case of the sensors, this bundle is remotely requested to provide the values of sensors and, in the case of the actuators, the X10 commands arrive to the correct appliance through this bundle.

The bundle guarantees that the command is valid and that its consequences check pre-established security policies (e.g. establishing the temperature of the air conditioning in some room to a dangerous value). R-OSGi is therefore the way of integrating each piece inside an OSGi-based component of the architecture, granting the communication between OSGi implementations.

The MAS is responsible for regularly checking the values of sensors, acting on the actuators accordingly (e.g. the temperature suddenly dropped, turn the heat on) and calling for help, as well as logging all the events and decisions. The aim is to make accessible the functionalities of an agent (e.g., its methods) as services. It would not be advisable to convert each agent into an OSGi bundle, since it would increase the development time and throw away the advantages of MAS based methodologies for problem solving. Therefore, the decision was to create an OSGi bundle that could make the bridge between regular bundles and Jade: the MAS bundle. This bundle can deal with one Agent Container (AC) and implement the methods declared in the interface of the agents in that AC as its own services. Moreover, this bundle must be able to start and stop agents, which, in practice, corresponds to the starting and stopping events of the services. The bundle, upon the reception of an invocation for an offered service from any other bundle, sends the invocation to the correspondent agent and delivers the respective result to the calling bundle. It must be noted that an agent, when trying to satisfy an invocation, may require the services provided by other bundles currently available. This is possible through the MAS bundle.Aa JadeGateway agent (JGa) is also used. The task of this agent is to act as a bridge between Jade and non-Jade code. This agent is created when the MAS bundle is started, along with the other agents. The JGa has the knowledge about which services are provided by each agent, whenever a request from a service arrives to the MAS bundle. Likewise, if an agent needs to use a service from another bundle, it contacts the MAS bundle, which is responsible for contacting the correct bundle, invoking the service and forwarding the result back to the agent. So, a bundle, which allows for Jad instances to run behind OSGi implementations, is created in a completely transparent way.

At a high level, the architecture is composed by components that share information based on Web Services. Each one of these components can then be detailed and looked closer in means of the pieces they are made of: sensors, actuators, MAS and software. The communication inside the components is based on OSGi and R-OSGi open standards, granting extensibility, modularity, dynamics and a logical
hierarchical organization of the pieces that support each component.

THE SIMULATION ENVIRONMENT

Before implementing such a complex architecture, it was advisable to create a simulation environment that allows for the system to be tested and assessed. In our case, we clearly need to study the behaviour of the system when specific cases occur, ranging from the reactive cases (e.g., react to a temperature change) to the more complex cases (e.g., there is no movement in the last room the person was spotted for a long period of time) (Carneiro and Novais 2008). We also want to know what the behaviour when malfunctioning of some components or in a multiple alarm scenario. Simulation enables the study of specific scenarios that usually do not occur but are possible, without facing hard consequences. We are therefore using simulation for studying the behaviour of the future house and making software improvements before buying equipments.

Our simulation environment is a house with four rooms, the environment around the house and the user itself. The first step was to select which parameters to simulate. We have already acquired the 1-Wire temperature and luminosity sensors and this data do not need to be simulated. We intended to simulate the movement and humidity sensors, the fire, flood and gas alarm, the user’s vital sign acquisition, (e.g., heard rate, the body temperature and the blood pressure) and an outside weather station. For actuators, we are simulating home appliances, ranging from an oven or a coffee machine to the lights or HiFi. The way these components are simulated is described in this section.

A major decision that we took was to develop all our simulation using OSGi and R-OSGi bundles. This means that the current architecture and logic organization of the components is the same that the final system. So, we expected to fasten the last phase of the development of the project since we will only need to replace the generation of the simulated data by the real components. The remaining system is the same and has already been hardly tested, which gives us great confidence on the performance of the final system. It is therefore clear that the simulation occurs only in the generation of data from the sensors and in the state of the appliances.

SIMULATING SENSORS

The temperature and luminosity sensors are already real 1-Wire sensors and do not need to be simulated. However, there is a group of sensors that are being simulated. There is an outside weather station that will be later replaced by a real 1-Wire pre-assembled one. This weather station (as the real one) can provide information about the wind speed and direction, temperature, humidity, barometric pressure, rain fall, sunlight intensity and lightening. In this way, we have access to the knowledge regarding the environment around the house. This is useful not only for informing the user but also to be accessed by the Group Decision Support System. The humidity sensors are used mainly in the bathroom, to detect that the user is having a shower and in the rest of the house, to monitor the air humidity. The TAI8540A 1-Wire Humidity Module will replace it. The fire, flood and gas sensors as well as the movement sensor, in other hand, are X10 based. The fire, flood and gas sensors are used to detect dangerous situations that threat the life of the user and the real ones will, respectively, be the PR8307, PR8306 and PR8808. The movement sensor is used to determine in which room the user is in and the real one will be the PR8070. As for the vital signs of the user, we intended to use the Vital Jacket developed in the University of Aveiro. However, by now, we are simulating the values of the heart rate, the body temperature and the blood pressure. This data are mainly remotely used by the Group Decision Support System and to raise alarms in case of abnormal readings.

The question is now how to generate the values. Should they be random? Should they follow a function? Should it be a mixture of both? We cannot forget that we also want to create specific scenarios. So, we need to find ways to both set the values manually and to let them evolve in a natural way. The way we choose to implement scenarios uses XML files. Each XML file starts with the address of the simulated sensor and is followed by a list of pairs of the type <tick, value>. Whenever a sensor bundle is started in the scenario mode, it searches for XML files that refer to any of the virtual sensors it controls. Let’s say that we are talking about a temperature bundle and it founds a XML file that refers to temperature sensors it controls. The bundle starts counting the seconds since it started and whenever the value of the seconds passed is equal to tick, the value of the simulated sensor becomes value. As an example:

```xml
<?xml version="1.0"?>
<scenario>
    <address>1AB2CD44200A1</address>
    <events>
        <value tick = "5">20</value>
        <value tick = "10">22</value>
        <value tick = "15">24.5</value>
    </events>
</scenario>
```

This XML file would set the temperature of the sensor to 20 degrees after 5 seconds after starting the bundle, to 22 degrees after 10 and to 24.5 degrees 15 seconds after starting the bundle, remaining like that until the bundle is stopped. The same methodology is used for any of the other sensors, being the only difference the values (that must be according to the type of sensor).
In the second case, we don’t want to set up static scenarios. Instead, we want to let the values flow and see what happens. Our decision here was not to study how values like luminosity or humidity would evolve during the day to implement it in the simulation. We decided to estimate a mean value for each of the sensors and create random values that are, however, inside the values that can be read from the real sensors (all of them have a working range). As an example, the temperature sensors we are using have its working range from -10°C to +85°C. The decision on generating random values for the sensors is due to the fact that thru this approximation we can see how our system reacts to oddest situations. Situations that we may have not predicted but that may occur in real life (due to special circumstances, like sensor anomalies). We concluded that if we pressed the temperature sensor with our hands (thus conducting electricity) the value of the temperature registered would always be 85°C. The system must be ready for such error events and by generating random values. We can see how it reacts to a wide range of different dispositions. However, we found out that by setting all the sensors randomly, the decisions would be very confuse. So, what we are doing is to set to random mode the sensors that we are studying at a given time and set the remaining sensors to normal values through those scenarios. This way, we can study the effects on the system in a more accurate and restrict way.

On the other hand, we can set the sensor bundles normal mode. In this mode, mathematical functions are used in an attempt to simulate the evolution of the values in real life in different situations. We can, as an example, simulate the temperature in a small or in a large room, or in a room that is more or less isolated from the exterior. Following this example, let’s see how we simulate the temperature sensors. We concluded that the temperature in a room depends on two main factors: the outside temperature and the temperature of the air conditioning. The way these parameters influence temperature depends on factors like the room size, the isolation and the number of windows or the number of air conditioning devices. To start, the outside temperature is provided by the simulated weather station. In its normal mode, it is generated by a Gaussian distribution which mean and variance can be changed according to what we want without, however, exceeding the working range of the temperature sensor. For the temperature sensor we consider yet a third factor: the previous temperature. We started using that when we realised that the temperature in a room does not change immediately when the outside or air conditioning temperature change. This way, we can simulate that delay. So, the function for the temperature in a given room becomes

\[ \text{temp}_t = 0.6 \times \text{ac}_{\text{temp}} + 0.2 \times \text{outside}_{\text{temp}} + 0.2 \times \text{temp}_{t-1} \]

If we wanted to simulate a room that has no walls or windows in contact with the exterior, or we were built with good isolating material, we would remove or decrease the outside\_temp parameter and increase it if it was a room with lots of windows. If we want to simulate a room with insufficient air conditioning systems, we would increase the temp\_t factor and decrease the ac\_temp.

As another example, we can look at the luminosity. The luminosity depends on three factors: the outside luminosity, the position of the window blinds and the fact of the lights being on or off. The values of the outside luminosity, in its normal mode, are also generated by a Gaussian function. The state of the lights and the window blinds is read from the X10 devices (see next section). So, if the lights are on, the luminosity level is always high. Otherwise, it depends on the existence of windows and, if they exist, the position of the window blinds. If they are down or no windows exist, the luminosity is low. Otherwise, the luminosity in a given room goes as follows:

\[ \text{luminosity} = \text{wb}_{\text{position}} \times \text{outside}_{\text{luminosity}} \]

where wb\_position is the percentage of the position of the window blind (e.g. a window blind that is at the middle of the window corresponds to wb\_position = 0.5). Similar processes occur for the other sensors. While the weather station and the vital signs are ruled by Gaussian functions that can be configured in runtime, the values of sensors inside the house depend on those external values and other factors (like house architecture).

A special case of simulation is the user movement simulation. In this case, we are using Markov Chains to
simulate the movement of the person. The state is the room where the person is and the transitions are the moves of persons walking from a room to another one. Consequently, there are no transitions between rooms that are not adjacent. The transitions have into account our habits. As an example, when we are in the hall, we have more probabilities to go to the living room than to the bathroom since we usually spend more time there.

Another important parameter in the simulation is the rate at which values are generated. While the heart rate can change very rapidly, the temperature usually does not change like that. So, for the vital signs are generated every two seconds while outside sensors are generated every minute. Between the sensors inside the house, there are also differences. The movement sensors change more frequently than the temperature or humidity ones.

To help us to understand those effects and to interact with the simulation platform, we have designed interfaces. In figure 6, we can see the interface, which shows the values of the sensors and alarms in the house. It also shows the vital signs of the user and the values of the outside weather station. Note that in this interface, only the values from the temperature and luminosity are obtained from real sensors. In figure 6, we can see the interface of the Sensor Manager. The sensor Manager allows us to attach or detach sensors to rooms, view the real 1-wire sensors in the network, view the description of the sensors as well as their functionalities, and the historic log. It also allows for the visualization of the values of the sensors in each room, and registers every operation done with the interface. This manager also shows alarms that fire when the temperature goes beyond some value or some sensor in use is not reachable in the real network.

SIMULATING ACTUATORS

The actuators in the house are responsible for controlling the appliances. They are X10 based. For test purposes, we have acquired one PR2025 which enables to control a light, one PR2060 which enables to control an electric appliance and a PR3130 which is a remote control to send X10 commands to the power line through the PR2060. However, with this equipment, we can only control one appliance and one lamp. The remaining equipments of the house must be simulated.

We therefore simulated the lights, the window blinds, a coffee machine, a HiFi, a television and the air conditioning system. The light simulation enables us to send commands to turn the lights on or off. The window blinds simulation enables us to pull the blinds up or down. The control of the coffee machine, HiFi and television are essentially used to turn them on or off according to the policies of the house (e.g. when the room is empty, at given hours). As for the air conditioning simulation, it allows us to turn it on, of, more cold and more hot. We also have to say that all of these virtual X10 actuators allow for their state to be checked i.e., we can read the position of the window blind or the actual temperature of the air conditioning system. We defined two ways of interacting with the virtual appliances. We can either generate specific scenarios or they can run in normal mode. In the scenarios, the methodology used is the same that was used for the sensors. We create a XML file for a specific X10 address which represents an appliance and that has a list of valid states. In the correct time, the state for that appliance is active. In the normal, the actuators behave much like they behave when controlled by a human. This means that their state will only change when told so by the system, unless there is some malfunctioning. Therefore, when the system sets the lights to be on, they will remain on until new contradictorily order.

CONCLUSION

In this paper, the architecture to support virtual-eCare is proposed. A simulation platform has been developed in order to improve and make some adjustments before the ultimate challenge: the implementation in one major Portuguese hospital.

REFERENCES


Dosi, G. 1998. “Sources, procedures and microeconomics effects of innovation”. In Economic Literature 26, 1120-1171.


BIOGRAPHY
DAVIDE CARNEIRO is a MSc student at the Department of Informatics, the University of Minho, Braga, Portugal. His current research directions span the fields of Ambient Intelligence, Collaborative Networks and Multi-agent Systems.

RICARDO COSTA is a Professor of Computer Science at the College of Management and Technology of Polytechnic of Porto, Felgueiras, Portugal. He is also a PhD student at the Department of Informatics at the University of Minho, Braga, Portugal. His current research directions span the fields of Ambient Intelligence, Collaborative Networks and Reasoning Systems applied to the Healthcare sector.

PAULO NOVAIS is a Professor of Computer Science at the Department of Informatics at the University of Minho, Braga, Portugal. He received a PhD in Computer Science from the same university in 2003. His current research directions span the fields of Knowledge Representation and Reasoning, Multi-Agent Systems, Ambient Intelligence, Collaborative Networks and AI and The Law. He is supervising several M.Sc. and Ph.D. project and he has published more than fifty papers in international journals, conferences and workshops and has co-edited two books. He is Vice-president of APPIA, the Portuguese Association for Artificial Intelligence.

JOSÉ MACHADO is Professor of Computer Science at the Department of Informatics, the University of Minho, Braga, Portugal. He received a PhD in Computer Science from the University of Minho, in 2002. His current research directions span the domains of Artificial Intelligence, Databases and Medical and Biomedical Informatics, with particular interests in the field of Knowledge Representation and Reasoning, Multiagent Systems, Ambient Intelligence, AI and The Law, AI in Medicine and Medical Records. He has published more than sixty papers in international journals, conferences and workshops and has co-edited two books.

JOSÉ NEVES is Full Professor of Computer Science at the Department of Informatics at the University of Minho, Braga, Portugal. He received a PhD in Computer Science from Herriot Watt University, Edinburgh, Scotland, in 1984. His current research directions span the fields of AI and The Law, Knowledge Representation and Reasoning and Multiagent Systems. He has published more than one hundred papers in international journals, conferences, workshops and two books.
A SIMULATION-BASED MODEL OF ABDUCTION

Gildas Morvan  Daniel Dupont  Philippe Kubiak
LGI2A - EA 3926  Département Ingénierie  LAGIS - UMR CNRS 8146
Université d’Artois  et sciences du vivant  École Centrale de Lille
Technoparc Futura  HEI, 13 rue Toul  Cité Scientifique - BP 48
F-62400 Béthune  F-59800 Lille  F-59651 Villeneuve d’Ascq
e-mail: gildas.morvan  email: daniel.dupont@hei.fr  email: philippe.kubiak@ig2i.fr
@fsa.univ-artois.fr

KEYWORDS
simulation-based reasoning, abduction, forensic entomology

ABSTRACT

Abduction, or Inference to the Best Explanation (IBE), is a reasoning process that generates possible explanations from a set of “surprising” observations. In this paper, a simulation-based model of abduction is introduced. This model is then implemented to develop a decision support system in the field of Forensic Entomology to help forensic scientists solve complex cases.

INTRODUCTION

Abduction, or Inference to the Best Explanation (IBE), is a reasoning process that generates possible explanations from a set of “surprising” observations. It has been widely studied in Philosophy and later on, in Artificial Intelligence (AI). The literature on this subject is so vast that a whole book would not be enough to undertake a complete survey. Therefore, only the key aspects of some of the abduction models developed in AI, set cover and logic based approaches, are briefly introduced. However readers may refer to Paul (1993) for a more comprehensive presentation.

Set cover based approaches define an abductive framework as a triplet \( < \Phi, \Omega, e > \) where \( \Phi \) is a set of hypotheses, \( \Omega \) is a set of manifestations, and \( e \) is a mapping from \( 2^\Phi \) to \( 2^\Omega \). The abductive problem is then defined as follows: Let \( \Omega^* \subseteq \Omega \) be a set of observed manifestations. The problem is to determine a minimal set \( \Phi^* \subseteq \Phi \) such as \( e(\Phi^*) = \Omega^* \).

Logic based approaches assume that domain knowledge is encapsulated in a theory \( T \) defined over a language \( \mathcal{L} \). Let \( \mathcal{A} \) be a set of sentences of \( \mathcal{L} \), be a set of hypotheses. Let the sentence \( \omega \) be a surprising observation (i.e., \( T \nvdash \omega \)). \( \phi \) is an explanation of \( \omega \) iff \( T \cup \phi \vdash \omega \), \( T \cup \phi \) is consistent and \( \phi \in \mathcal{A} \).

Existing abduction AI models rely on knowledge models structured a specific language, defining truth values for propositions or relations between facts. However, the knowledge about a given system can be encapsulated in many types of models, e.g., a computational model. The abduction model presented in this paper does not assume an \( a \ priori \) knowledge of causality relations nor a specific syntax for the knowledge base, but assumes that: (1) the set of possible explanations (hypotheses) can be defined as a metric space — i.e., distance measures can be computed between all the hypotheses; (2) it exists a deductive (predictive) model able, e.g., by simulating the causal history of the system, to compute a coherence measure between a hypothesis and a set of observations. Then, this model is applied to develop a decision support system in the field of Forensic Entomology, using agent-based simulations. Experimental results show that using this model to solve complex cases increases the efficiency and the precision of the results.

A SIMULATION-BASED MODEL OF ABDUCTION

Definitions

Let \( \Omega \) be a set of observations and \( \Phi \) a set of hypotheses — i.e., possible explanations — for \( \Omega \). We assume that a metric space \( (\Phi, d^\Phi) \) can be defined. A coherence measure between an hypothesis \( \phi \) of \( \Phi \) and \( \Omega \) is computed by a deductive (predictive) model \( m : \Phi \times \Omega \rightarrow [0,1] \). This coherence measure is denoted \( c_{m,\phi,\Omega} \).

**Definition 1** A hypothesis is called validated iff its coherence measure is known. Let \( \Phi^c \subseteq \Phi \) be the set of validated hypotheses and \( R_{m,\Phi^c,\Omega} : \Phi \times [0,1] \) be the solution set computed by a model \( m \):

\[
R_{m,\Phi^c,\Omega} = \bigcup_{\forall \phi \in \Phi^c} (\phi, c_{m,\phi,\Omega}).
\] (1)

Similarly, a metric space \( (R_{m,\Phi^c,\Omega}, d^R) \) is defined. An estimated coherence measure \( \hat{c}_{m,\phi,\Omega} \) is associated to all \( \phi \in \Phi - \Phi^c \). This measure is estimated from \( R_{m,\Phi^c,\Omega} \) by interpolation. The resulting estimated solution set is defined as follows:

\[
\hat{R}_{m,\Phi,\Omega} = \bigcup_{\forall \phi \in \Phi} \begin{cases} 
(\phi, c_{m,\phi,\Omega}) & \text{if } \phi \in \Phi^c \\
(\phi, \hat{c}_{m,\phi,\Omega}) & \text{otherwise.}
\end{cases}
\] (2)
Table 1: Goals of an abductive task

<table>
<thead>
<tr>
<th>Goal</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>pr.</td>
<td>Determine the most probable hypotheses</td>
</tr>
<tr>
<td>el.</td>
<td>Eliminate improbable hypotheses</td>
</tr>
<tr>
<td>prs.</td>
<td>Determine ( c_{m,\phi,\Omega} ) for probable hypotheses</td>
</tr>
<tr>
<td>id.</td>
<td>Determine ( c_{m,\phi,\Omega} ) for every hypotheses</td>
</tr>
</tbody>
</table>

Abduction goals

It is generally accepted that abduction targets the best explanation of a given set of observations. However, in the context of a decision support system, this definition should be expanded. In the table 1, four different goals for an abductive task are defined. Thus, in its most general definition, abduction can be seen as an identification task (goal id.).

Quality measures can be defined to determine how goals have been achieved. A quality measure, \( q_{m,\Omega} \), based on the root mean square deviation (RMSD) between \( R_{m,\Phi^e,\Omega} \) and \( R_{m,\Phi,\Omega} \) is defined for each goal \( g \) of the table 1. \( \text{E.g., for the goal id. the quality measure is defined as follows:} \)

\[
q_{m,\Omega} = \sqrt{\frac{\sum_{\phi \in R_{m,\Phi,\Omega}} (c_{m,\phi,\Omega} - c_{m,\phi^e,\Omega})^2}{|\Phi|}}.
\]

For the other goals, subsets of \( R_{m,\Phi,\Omega} \) and \( R_{m,\Phi^e,\Omega} \), meeting their defining condition — \( \text{e.g., for the goal prs.} \) subsets are defined such as, for any \( \phi \) in \( R_{m,\Phi,\Omega} \), \( c_{m,\phi,\Omega} > 0 \) — are used.

Knowing the result set is mandatory to use these quality measures. Thus, they are useful to evaluate an implementation of the algorithm — and especially the heuristic \( \gamma \) —, not a particular solution.

Resolution algorithm

The algorithm 1 computes an estimated solution set \( R_{m,\Phi^e,\Omega} \) using a deductive model \( m \). One could argue that it is not an abductive algorithm as it does not return the best explanations for \( \Omega \). Nevertheless, as it explains possible causal histories of \( \Omega \), it is, in some ways, similar to the abductive framework developed in Lipton (2004; ch. 4). Moreover, the solution set can be used as an input for post-processing algorithms, \( \text{e.g., to merge solution sets obtained with different deductive models, and then determine the best explanation. Halting of the algorithm 1 depends on a condition C. In the implementation presented in the next section, the following condition is used:} \)

\[
\sum_{r \in R_{m,\Phi^e,\Omega}} \text{cost}(r) \leq \max \text{Cost}, \tag{5}
\]

where \( \text{cost} \) represents the cost necessary to select and validate an hypothesis and \( \max \text{Cost} \) is the maximal cost allowed by the user to compute the solution. Of course the algorithm halts if all the \( \phi \) in \( \Phi \) have been validated. Using a condition of the type

\[
c_{m,\phi,\Omega} \geq \mu, \tag{6}
\]

where \( \mu \) is a validity threshold, the algorithm would be a form of the hypothetico-deductive model (Lipton 2004; ch. 4).

**Algorithm 1** Inquisitive abduction algorithm

**Input:** A set of observations \( \Omega \)
**Input:** A halting condition \( C \)
**Input:** A heuristic \( \gamma \)
**Output:** A solution set \( R_{m,\Phi^e,\Omega} \)

1. Determine \( \Phi \)
2. \( \Phi^e \leftarrow \emptyset \)
3. \( R_{m,\Phi^e,\Omega} \leftarrow \emptyset \)
4. compute \( R_{m,\Phi,\Omega} \)
5. while not \((C \text{ or } (\Phi = \Phi^e))\)
6. \( \phi \leftarrow \gamma(R_{m,\Phi,\Omega}) \)
7. \( c_{m,\phi,\Omega} \leftarrow m(\phi, \Omega) \)
8. \( \Phi^e \leftarrow \Phi^e \cup \phi \)
9. \( R_{m,\Phi^e,\Omega} \leftarrow R_{m,\Phi^e,\Omega} \cup (\phi, c_{m,\phi,\Omega}) \)
10. compute \( R_{m,\Phi,\Omega} \)
11. end while
12. return \( R_{m,\Phi^e,\Omega} \)

In the worst case, \( i.e., \) the halting condition \( C \) stays false during the entire execution, this algorithm needs \( |\Phi| \) steps to halt and return \( R_{m,\Phi,\Omega} \), \( i.e., \) the exact solution set. However, the complexity of the algorithm depends mainly on the complexity of the heuristic and on the solution set estimation algorithm. This algorithm is very general; thus, it does not refer to the goal of the abductive task; this is handled by the heuristic \( \gamma \). In the next section, elements for constructing such heuristics are presented. We focus on the goals \( pr., \text{ prs. and id.:} \) for the goal \( pr. \), classical optimisation heuristics can be used efficiently. Moreover, we target cases where the hypothesis set is not very big but the cost of validation of a hypothesis is important.

**HEURISTIC DEFINITIONS**

**Definition of the criteria**

Many criteria that help an agent to choose the best hypothesis to validate have been identified in the literature. Among all of these, simplicity seems to have been
the most used by computer scientists. Simplicity has been interpreted as logical simplicity. But it seems obvious that for a cognitive agent, simplicity is a little more complex (Aliseda-Llera 1998, Paul 1993). In the model presented in this paper, simplicity is defined as follows:

**Definition 2** A hypothesis $\phi$ is simple if it exists a retrodictive model $m^{-1}$ such as $m^{-1}(\Omega) = \phi$. Moreover, if $\Phi$ is bounded, $\phi$ should be considered as simple if $\phi = \inf(\Phi)$ or $\phi = \sup(\Phi)$.

The simplicity criterion is used to preprocess the problem. Thus, all the hypotheses defined as simple will be primarily chosen and validated.

Other criteria such as cost or utility have been quoted and should be used to handle the choice of a hypothesis in an abductive task (Peirce 1931, McGrew 2003). The cost criterion can be defined informally as follows: the selected hypothesis should be, ceteris paribus, the one that minimises the cost of validation. This criterion should be used if it is possible to estimate empirically a cost function of the model $m$, $\text{cost}_m : \Phi \rightarrow \mathbb{R}$.

Let $c : \Phi \rightarrow [0, 1]$ be a utility function of the cost criterion. For any $\phi \in \Phi$

$$u_c(\phi) = 1 - \frac{\text{cost}_m(\phi)}{\max(\text{cost}_m)}.$$  \hspace{1cm} (7)

The utility criterion can be interpreted in different ways. In this approach, it could be defined as follows: the selected hypothesis should be, ceteris paribus, the one which maximises the knowledge of the agent. It means that the chosen hypothesis should be the one which best improves the quality of the solution — relatively to a quality measure. Of course in a real world problem, such an hypothesis as well as its effect on the quality measure cannot be determined with certainty. However, the metric of $\Phi$ or $\mathcal{R}_{m, \Phi, \Omega}$ can be used to estimate a utility function for this criterion. First, consecutivity is defined:

**Definition 3** Let $(E, d)$ be a metric space. Two elements $e_i$ and $e_j$ of $E$ are consecutive in $(E, d)$ if it does not exist an element $e$ of $E$ such as

$$d(e, e_i) < d(e, e_j) \text{ and } d(e, e_j) < d(e, e_i).$$  \hspace{1cm} (8)

The notion of maximal distance between two consecutive elements in a metric space is then defined as follows:

**Definition 4** Let $(E, d)$ be a metric space and $d_{\text{max}}((E, d))$ the maximal distance between two consecutive elements in $(E, d)$. For any consecutive elements $e_i$ and $e_j$ of $E$, $d_{\text{max}}((E, d)) = d(e_i, e_j)$ if it does not exist two consecutive elements $e_k$ and $e_l$ in $(E, d)$ such as $d(e_k, e_l) > d(e_i, e_j)$.

Two different utility functions of the utility criterion can then be defined using the metrics of $\Phi$ — equation 9 — and $\mathcal{R}_{m, \Phi, \Omega}$ — equation 10. Let $u^\Phi, u^R : \Phi \rightarrow [0, 1]$ be two utility functions of the utility criterion. For any $\phi$ of $\Phi$

$$u^\Phi(\phi) = \begin{cases} 0 & \text{if } \phi \in \Phi^v \\ \frac{d^\Phi(\phi, \phi_1)}{d_{\text{max}}(\Phi^v)} & \text{otherwise} \end{cases} \hspace{1cm} (9)$$

where the hypothesis $\phi_1 \neq \phi$ is the closest to $\phi$ in $\Phi^v \cup \{\phi\}$, i.e., there is no $\phi_2 \neq \phi$ in $\Phi^v \cup \{\phi\}$ such as $d^\Phi(\phi, \phi_2) < d^\Phi(\phi, \phi_1)$.

For any $r = (\phi, c_m, \phi_1) \in \mathcal{R}_{m, \Phi, \Omega}$, $c_m, \phi_1$ may be unknown

$$u^R(\phi) = \begin{cases} 0 & \text{if } \phi \in \Phi^v \\ \frac{d^R(\phi, \phi_1)}{d_{\text{max}}(\mathcal{R}_{m, \Phi, \Omega})} & \text{otherwise} \end{cases} \hspace{1cm} (10)$$

where the result $r_1 \neq r$ is the closest to $r$ in $\mathcal{R}_{m, \Phi, \Omega} \cup \{r\}$, i.e., there is no $r_2 \neq r$ in $\mathcal{R}_{m, \Phi, \Omega} \cup \{r\}$ such as $d^R(\phi, r_2) < d^R(\phi, r_1)$.

For any $\phi$ of $\Phi$, if $\phi$ belongs to $\Phi^v$, $u^\Phi(\phi) = u^R(\phi) = 0$.

This is a desired property as it is useless to validate a hypothesis twice with the same deductive model.

**Heuristic definitions**

An aggregation operator is used to determine the global utility functions $u^\Phi$ and $u^R$ — using respectively $u^\Phi$ and $u^R$ — of a hypothesis. As an example, weighted sum is used here. However any aggregation operator can also be used.

Let $u^\Phi, u^R : \Phi \rightarrow [0, 1]$ be two functions representing the utility of any hypothesis of $\Phi$. For any $\phi$ of $\Phi$

$$u^\Phi(\phi) = \alpha_\Phi \cdot u_c(\phi) + \beta_\Phi \cdot u^\Phi(\phi),$$  \hspace{1cm} (11)

with $\alpha_\Phi + \beta_\Phi = 1$.

For any $r = (\phi, c_m, \phi_1) \in \mathcal{R}_{m, \Phi, \Omega}$

$$u_R(\phi) = \alpha_R \cdot u_c(\phi) + \beta_R \cdot u^R(\phi),$$  \hspace{1cm} (12)

with $\alpha_R + \beta_R = 1$.

Two heuristics $\gamma^\phi$ and $\gamma^R$ can now be defined.

**Definition 5** Let $\gamma^\phi$ (resp. $\gamma^R$) : $\mathcal{R}_{m, \Phi, \Omega} \rightarrow \Phi - \Phi^v$ be a function that chooses an hypothesis to be validated. For any $\gamma^R$ = $(\phi, c_m, \Phi, \Omega) \in \mathcal{R}_{m, \Phi, \Omega}$, $\gamma^\phi(\mathcal{R}_{m, \Phi, \Omega})$ (resp. $\gamma^R(\mathcal{R}_{m, \Phi, \Omega})$) = $r$ if it does not exist a $r'$ = $(\phi', c_m, \Phi, \Omega)$ in $\mathcal{R}_{m, \Phi, \Omega}$ such as $u^\phi(\phi') > u^R(\phi)$ (resp. $u^R(\phi') > u^R(\phi)$) and $\phi$ does not belong to $\Phi^v$. The efficiency of these heuristics is evaluated on a real world problem presented in the following section. The goals $\text{cl. et id.}$ are considered.

**APPLICATION TO FORENSIC ENTOMOLOGY**

**Introduction to Forensic Entomology**

Forensic entomology is widely used in criminal investigations to determine post-mortem intervals (PMI) from...
the insects found on a cadaver. A PMI is usually estimated by experts using retrodictive models. These models are very easy to use but they do not take into account the ecosystemic context; thus, estimations performed using these methods are often overestimated and not as precise as they could be. Modern PMI estimation methods are based on insect development models. These models consider that insect development speed is temperature-depandent (Stinner et al. 1974). It is given as a function \( f \) of the temperature \( T \) varying in the time \( t \). When a cadaver is discovered, investigators take insect samples from the body. Entomologists determine the species and the accumulated rates of development, denoted \( \Delta a \), of the oldest individuals. Then, for each one of them, the laying time \( t_1 \) (generally close to the time of death) can be calculated from the following equation

\[
\Delta a = \int_{t_1}^{t_2} f(T(t)) dt,
\]

where \( t_2 \) represents the time of the cadaver discovery. Data from the nearest meteorological station are usually used in order to estimate \( T(t) \). However, it is problematic for many reasons. First, one can notice that cadavers are rarely found at the foot of a meteorological station. Furthermore, corpse thermal inertia is important, especially in the first hours after death. Finally the heat generated by larva aggregates can raise the temperature locally up to \( 20^\circ \)C. Thus, in many cases, entomological expertise results are inaccurate and given with an important margin of error.

**A decision support system for Forensic Entomology**

To handle these issues, a predictive agent-based model of insect development and cadaver decomposition in a complex ecosystem has been developed. This model is used to determine if a hypothesis — a possible time of death — is coherent with the observations available on the ecosystem of the crime scene and the entomofauna found on the victim. More information about this model and the validation process can be found in Morvan et al. (2007). As the model is stochastic, it is necessary to run a large number — about 100 in most of the cases — of simulations to compute a coherence measure statistically significant. For any \( \phi \) of \( \Phi \)

\[
\text{cost}_m(\phi_i) = \lambda \cdot \left(1 - \frac{i}{|\Phi| - 1}\right),
\]

where \( \lambda \) represents the validation cost of the hypothesis \( \phi_0 \).

Here we assume that \( \lambda = 1 \). Thus, 50.25 would be necessary to validate all the hypotheses of \( \Phi \). The maximal cost allowed to compute the solution is arbitrary fixed to 20. 10000 result sets have been generated to determine the best affection for \( \alpha_\phi \) and \( \alpha_R \). Results show that whatever the goal is \( \alpha_\phi \approx \alpha_R \approx 0.001 \). An interesting feature of this result is that the quality of the solution is very bad if the cost criterion is not used — i.e., if \( \alpha_\phi = \alpha_R = 0 \) — and best for very small values of \( \alpha_\phi \) and \( \alpha_R \). Results presented in the figures 1 and 2 show that \( \gamma_\phi \) and \( \gamma_R \) perform well respectively for the goals \( id \), and \( el \).

Many different development models of Diptera can be found in the literature. It has been shown in Wagner et al. (1984) that it is impossible, in the general case, to establish that a model is better than an other. Thus, the
multiagent model can be parametrised to use any development model. An experiment has been conducted on a real case using six different development models. Estimated result sets were merged using the Transferable Belief Model (Smets and Kennes 1994). The figure 3 shows the result of the merging after pignistic transformation. It allows to conclude that the victim died between $\phi_{55}$ and $\phi_{91}$, and that the most probable time of death is $\phi_{78}$.

This result is very interesting for at least two reasons. First, it shows that this new method allows to determine the most probable time of death more precisely than traditional methods, the agent-based model including more parameters than the classic analytic models. A discussion on the simulation results and their impact on the accuracy of PMI estimations can be found in Morvan et al. (2007). Second, a confidence interval can be formally determined from simulations results whereas in traditional methods the confidence interval estimation relies on the expert’s intuition and is often questionable. As the concept of proof is crucial in criminal investigations, such an improvement is particularly interesting.

CONCLUSION

In this paper, a model of abduction is introduced and heuristics are defined to implement efficiently this model. This model is then applied in the context of Forensic Entomology. However, hypothesis sets are unidimensional in this case and large experiments should be carried on multidimensional sets. But results are encouraging and the model is being applied to develop a decision support system for cervical cancer prevention. Moreover, such a model could be used to perform intelligent exploration of model predictions or solve complex inverse problems. More generally, this study attempts to show that simulation-based reasoning models can be very useful to produce inferences in the framework of complex systems that cannot be described in logic-based languages. It seems particularly true for abductive and inductive inferences that need a higher level description than deductive (predictive) inference.

REFERENCES


NEURAL NETWORKS AND AI TECHNIQUES
GRAMMATICAL EVOLUTION FOR DEVELOPMENT OF NEURAL NETWORKS WITH REAL-VALUED WEIGHTS USING CELLULAR ENCODING

Jan Drchal
Miroslav Šnorek
Czech Technical University, Faculty of Electrical Engineering,
Department of Computer Science and Engineering
Karlovo náměstí 13,
121 35, Prague, Czech Republic
email: {drchaj1|snorek}@fel.cvut.cz

KEYWORDS
Neural networks, Grammatical Evolution, Cellular Encoding

ABSTRACT

Artificial Neural Network is a well-known tool used for data modeling of systems. This paper focuses on so-called TWEANNs (Toplogy and Weight Evolving Artificial Neural Networks). TWEANNs are Evolutionary Algorithms (EAs) which evolve both topology and parameters (weights) of neural networks. Here, we concentrate on a use of an indirect developmental encoding which is an approach inspired by multi-cellular organisms’ development from a single cell (zygote) known from Nature. We examine multiple modifications of a known tree-based indirect developmental encoding: the Cellular Encoding, Grammatical Evolution (GE) is employed instead of Genetic Programming (GP) to optimize program trees. GE is advantageous mainly in the way it can handle constraints (as it evolves program trees which conform to a grammar specified using a BNF notation). Moreover, we employ GE’s inner mechanisms to efficiently encode Neural Network parameters (weights and biases). The Cellular Encoding is a neuron-centric approach, therefore an increased attention should be paid to a way in which a proper synaptic link is selected (prior to modification of its parameters). In this work, we compare three different link selection schemes. The results of our investigations show that our modifications of Cellular Encoding improve the ability to evolve real-valued Artificial Neural Networks.

INTRODUCTION

Using Evolutionary Algorithms (EAs) (Michalewicz 1994) to learn Artificial Neural Networks (ANNs) (Tsoukalas and Uhrig 1996) is a well examined approach as EAs are very robust. Even harder optimization problems must be solved for TWEANNs (Topology and Weight Evolving Artificial Neural Networks) – these algorithms are not limited to search for proper weight settings only but they also optimize the topologies of neural networks. TWEANN approach is useful for ANN user, who does not have to experiment with different topologies. Also finding (at least near) optimal topology leads to better data modeling results. The problem of most recent TWEANN algorithms is in their inability to evolve large-scale modular ANNs. This is mainly caused by the so-called curse of dimensionality, where optimization methods fail because of high-dimensional space to search through. Most current TWEANN methods use direct encoding approaches to represent ANNs. In direct encoding a single gene describes a single feature (e.g., a connection or a neuron) of an ANN. This is unlike in Nature where genome rather represents a ”program” to build the target organism. Human genome consists of roughly 30 000 genes and is able to encode more than 20 billion neurons, each linked to as many as 10 000 others, plus, of course, the rest of the organism. This efficient information storage can be seen as a kind of compression. Hence, the translation of a genotype to phenotype is a decomposition process. Of course, such a high level of compression is only possible, if the information is highly regular. This is true for Nature, as organisms and their brains are known to be highly modular, hierarchal systems. Artificial encodings which are trying to possess such attributes are known as indirect encodings.

The indirect developmental encodings for ANNs can be found for example in the works of Eggemberger (Eggenberger-Hotz 1997, Eggenberger-Hotz et al. 2003) or Stanley (D’Ambrosio and Stanley 2007, Gauci and Stanley 2007).

This work particularly focuses on experiments with tree-based indirect encodings of ANNs. In (Gruau 1994) Gruau introduced an indirect encoding called Cellular Encoding (CE), where the development of ANNs is controlled by evolved program trees. Trees are well examined data structures and there was already a great amount of interest in development of tree structure optimization Evolutionary Algorithms. The most widely known, Koza’s Genetic Programming (GP) (Koza 1992), which was originally used to evolve
LISP program trees, was adopted by Gruau and used to optimize CE development trees. Here, we employ GP’s successor Grammatical Evolution (GE) as we have already used it in (Drlach and Šnørek 2008). In contrast with the previous work, where we have experimented with boolean neural networks, here, we compare different modifications of CE which are able to encode ANNs with real-valued weights.

**EVOLUTION OF TREE STRUCTURES**

Koza’s Genetic Programming (GP) (Koza 1992) is a well-known Evolutionary Algorithm for optimization of tree structures. GP trees can represent mathematical models of data or programs. Figure shows a simple function of two variables described by an evolved GP tree. The tree nodes are labeled by symbols which can represent either operations, variables or constants. Each node has a predefined arity (number of child nodes). Constants and variables are localized in leaf nodes. GP was already used for evolution of ANNs (Koza and Rice 1991). However, the encoding was direct.

![Genetic Programming evolved tree](image)

Figure 1: Genetic Programming evolved tree. This tree is assembled of 7 symbols, it uses three operations (+, *, /), two variables (x, y) and two constants (2, 5). It represents a function \( f(x, y) = (x \times 2) + (5/y). \)

**GRAMMATICAL EVOLUTION**

In (Ryan et al. 1998) Ryan, Collins and O’Neil introduced a Grammatical Evolution (GE) which is an Evolutionary Algorithm able to evolve solutions according to a user-specified grammar. Unlike GP, GE brings more flexibility as user is able to constrain the way in which the program symbols are assembled together. In GP it is impossible to control their order – any symbol can become a child or a parent of any other symbol. The GE approach is therefore able to radically cut down the search space. The grammar in GE is specified using Backus Naur Form (BNF). GE uses linear genome of integer numbers (codons). When expanding a non-terminal symbol having a choice from \( n \) possible rewriting rules, we chose rule No. \( iMODn \), where \( i \) is the integer value of the actual codon. The problem arises when we need to encode a real-valued constants, most often a grammar is expanded to do so (?). The other approach is to use the integer number which encodes a codon and transform it to right interval ( ).

**CELLULAR ENCODING**

Cellular Encoding (CE) was introduced by Gruau in Gruau (1994), where more detailed description can be found. In CE, ANN development starts with a single cell (neuron). A development tree is traversed using breath-first search starting from the root node. CE uses symbols for cell division (PAR, SEQ) and updating of cell registers like neuron bias or input link pointer (B+, INC etc.). The original Cellular Encoding was limited to Boolean Neural Networks where network’s input and output is binary. Boolean Neural Networks use neurons which fire 1 for activities above given threshold and 0 for lower. Weights can be either 1 or -1. The following list contains short description of symbols used in CE development trees:

- **SEQ** sequential division – a new cell inherits mother cell’s outputs, input of a new cell is connected to the mother cell’s output using a connection of weight 1. Development instructions for a mother cell continues in a left subtree, while for a new cell in the right.
- **PAR** parallel division – create a new cell which has the same set of inputs and outputs as the mother cell.
- **INC, DEC** increase/decrease internal input link pointer.
- **W+, W−** set weight of an input link designated by an internal input link pointer to 1 or -1.
- **B+, B−** increase/decrease bias by 1.
- **CUT** cut incoming connection given by the internal input link pointer.
- **WAIT** do nothing – continue with the next step (needed for synchronization).
- **END** end processing of this cell.

Gruau has later developed a modified method which is capable to encode real-valued weights Gruau et al. (1996).

**TESTED GRAMMARS**

This section describes CE based grammars which we have used in our experiments. Rather than using Gruau’s real-valued weight encoding Gruau et al. (1996) which was not suitable for Grammatical Encoding we have decided to modify the existing CE for Boolean Networks. The distinction is that we do not use any kind of special input and output cells (neurons). In the start, an initial cell (zygote) is fully connected to both
input neurons and output neurons. This modification led us to introduction of a new cell register the output link pointer which is an exact counterpart of the input link pointer mentioned above. It selects the neuron output link for link operations such as setting weight or cutting link. The CE had to be of course expanded by counterparts of INC, DEC and all symbols which manipulate links (\(\oplus\), \(\ominus\) and \(\text{CUT}\)). These changes to the original CE seems only to increase the dimensionality of the problem. However, we found, that it was impossible to evolve real-valued ANNs for even the simplest tasks without ability to control both input and output neuron’s links. This would be addressed in further research. Note, that we have omitted the \(\text{CUT}\) symbol in all of our grammars as it was not needed by the test problem (see the next Section). It can be simply added, though.

**PNT-GRA grammar**

The grammar designated PNT-GRA (link pointer-based, grammar encoded parameters) resembles the original CE described in the previous section (note that program symbols are shortened, this is in order to save space).

\[
\langle T \rangle ::= N|n|T>|S<T>|T>|P<T>|T>|I<T>|i<T>|  
O<T>|o<T>|T>|w<N>|T>|W<n>|T>|B<n>|T>
\]

\[
\langle N \rangle ::= <D><D><D>
\]

\[
\langle D \rangle ::= 0|1|2|3|4|5|6|7|8|9
\]

The non-terminal \(\langle T \rangle\) which represents a program tree is also a starting symbol. The symbol \(N\) corresponds to END, \(n\) to WAIT, \(S\) to SEQ and \(P\) to PAR. The symbols \(I, i, 0\) and \(o\) increase/decrease the cell’s input/output link pointer. The symbols \(W\) and \(W\) sets the proper input/output weight value and in the similar way the symbol \(B\) sets the cell’s bias to \(<N>\). Note, that the non-terminal \(<N>\) produces integers in a range from 0 to 999. In a final network this integer is linearly transformed to an interval \(-100..100\) for both weights and biases.

**PNT-INT**

The PNT-INT (link pointer-based, integer encoded parameters) grammar uses a different way to encode constants:

\[
\langle T \rangle ::= N|n|T>|S<T>|T>|P<T>|T>|I<T>|i<T>|  
O<T>|o<T>|T>|w<N>|T>|W<n>|T>|B<n>|T>
\]

\[
\langle I \rangle ::= C|c
\]

Here, the non-terminal \(\langle T \rangle\) was replaced by a non-terminal \(\langle I \rangle\) which can be rewritten to two special symbols \(C\) and \(c\). The codon value to rewrite \(\langle I \rangle\) is used as a parameter value. It is clear, that we have to use at least two symbols (\(C\) and \(c\)) – in a case of only a single rewriting rule no codon is used. To match the grammar encoded parameter approach (GRA) the used codons were also represented by integers from 0 to 999. Note that in comparison with the GRA approach which needs three codons to encode a parameter with the given precision, INT suffices with the only one.

**ABS-GRA**

The ABS-GRA (absolute link pointers, grammar encodes parameters) grammar is based on a PNT-GRA grammar.

\[
\langle T \rangle ::= N|n|T>|S<T>|T>|P<T>|T>|I<T>|i<T>|  
w<T>|<N>|T>|B<n>|T>
\]

\[
\langle I \rangle ::= C|c
\]

\[
\langle N \rangle ::= <D><D><D>
\]

\[
\langle D \rangle ::= 0|1|2|3|4|5|6|7|8|9
\]

Both link input pointer and link output pointer were removed. On the other hand the link modification symbols (\(W\) and \(W\)) were added an integer input \(\langle T \rangle\). The input/output link is selected as a modulo of a current number of input/output links and the integer.

**ABS-INT**

The grammar ABS-INT (absolute link pointers, integer encoded parameters) further simplifies the previous grammar by the use of integer parameter encoding:

\[
\langle T \rangle ::= N|n|T>|S<T>|T>|P<T>|T>|I<T>|i<T>|  
w<T>|<T>|B<n>|T>
\]

\[
\langle I \rangle ::= C|c
\]

**REL-INT**

The grammar REL-INT (relative link pointers, integer encoded parameters) is an attempt to simplify the classical PNT link pointer approach:

\[
\langle T \rangle ::= N|n|T>|S<T>|T>|P<T>|T>|Y|<T>|Y
\]

\[
\langle I \rangle ::= C|c
\]

In the classical approach, symbols \(I, i, 0\) and \(o\) increase/decrease the relevant link pointers. When the number of the neuron’s input/output link’s is high this leads to program trees with a high number of repeated symbols. REL approach introduces symbols \(Y\) and \(y\) which change input/output link pointers by an integer number from interval \(-5..5\).

**EXPERIMENTS**

In this section we show results obtained by using grammars introduced in the previous section. We have evolved ANNs with real-valued weights which solve a well-known XOR problem. XOR problem, although very simple, is a satisfactory tool to show the different behaviour of proposed encodings. The XOR function can be described by the Table 1:
Table 1: The XOR function table. The $x_1^i$, $x_2^i$ are the function parameters, while $y^i$ designates the function value.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$x_1^i$</th>
<th>$x_2^i$</th>
<th>$y^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The fitness $f$ of the developed ANNs $N(x_1^i, x_2^i)$ was computed using the following equation:

$$f(N) = 4.0 - \sum_{i=1}^{4} |N(x_1^i, x_2^i) - y^i|$$  \hspace{1cm} (1)

The network was considered successful when its fitness $f(N) \geq 3.9$. All experiments were done using population of 1500 individuals, with initial chromosome size of 20, crossover probability 0.9 and mutation probability 0.1. Wrapping was turned off. We have used a tournament selection with tournament size 2.

The averaged results over 1000 runs are shown in Tables 2 and 3. The Table 2 summarizes the grammatical approach of parameter encoding (PNT-GRA and ABS-GRA). We can see that the ABS link selection performs about 10% better in number of generations needed for optimization than the classical PNT approach. Also, the length of resulting ANN encoding is shorter by 8% on average.

Table 3 summarizes the integer approach (PNT-INT, ABS-INT and REL-INT). It is clear, that for both PNT and ABS link selection schemes the integer approach outperforms the grammatical one. The ABS-INT is the winner of all tested grammars. It beats the classic Gruau-like PNT-GRA by about 33% percent in the number of generations and 23% in the number of symbols needed. The REL scheme was the least successful of all grammars using the integer approach.

**CONCLUSIONS**

In this paper, several modifications of Cellular Encoding (CE) for real-weighted Artificial Neural Networks (ANNS) are presented. The modifications took advantage of the Grammatical Evolution (GE) algorithm used for evolving CE program-trees. The comparison was done in two ways:

- Comparison of different approaches of parameter encoding (weights and biases). We have compared a classical grammar approach (GRA) and an integer approach (INT) which benefits from inner mechanism of GE algorithm, where genome consist of integer codons.

Table 2: A comparison of link selection schemes PNT and ABS for grammatical approach of parameter encoding – grammars PNT-GRA and ABS-GRA. Table shows average (avg), standard deviation (stdv), minimum (min) and maximum (max) values for both number of generations needed for optimization and number of used symbols of evolved program trees.

<table>
<thead>
<tr>
<th>Generations</th>
<th>PNT</th>
<th>ABS</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>212.26</td>
<td>192.03</td>
</tr>
<tr>
<td>stdv</td>
<td>34.73</td>
<td>35.11</td>
</tr>
<tr>
<td>min</td>
<td>123</td>
<td>59</td>
</tr>
<tr>
<td>max</td>
<td>307</td>
<td>285</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbols</th>
<th>PNT</th>
<th>ABS</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>75.44</td>
<td>69.29</td>
</tr>
<tr>
<td>stdv</td>
<td>33.89</td>
<td>29.44</td>
</tr>
<tr>
<td>min</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>max</td>
<td>303</td>
<td>293</td>
</tr>
</tbody>
</table>

Table 3: A comparison of link selection schemes PNT, ABS and REL for integer approach of parameter encoding – grammars PNT-INT, ABS-INT and REL-INT.

<table>
<thead>
<tr>
<th>Generations</th>
<th>PNT</th>
<th>ABS</th>
<th>REL</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>148.81</td>
<td>133.04</td>
<td>210.06</td>
</tr>
<tr>
<td>stdv</td>
<td>25.79</td>
<td>29.32</td>
<td>34.04</td>
</tr>
<tr>
<td>min</td>
<td>48</td>
<td>39</td>
<td>3</td>
</tr>
<tr>
<td>max</td>
<td>216</td>
<td>216</td>
<td>319</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbols</th>
<th>PNT</th>
<th>ABS</th>
<th>REL</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>57.82</td>
<td>51.67</td>
<td>58.52</td>
</tr>
<tr>
<td>stdv</td>
<td>25.00</td>
<td>22.47</td>
<td>25.63</td>
</tr>
<tr>
<td>min</td>
<td>18</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>max</td>
<td>182</td>
<td>172</td>
<td>213</td>
</tr>
</tbody>
</table>

- Comparison of different link selection schemes: a classical pointer to link scheme (PNT), our absolute selection scheme (ABS) and our modification of a PNT scheme – the relative selection scheme (REL).

The experiments shown the superiority of INT over GRA parameter encoding approach. This can be explained by the fact that the INT is able to encode shorter genomes (in our case only a single symbol was needed instead of three to encode a parameter). Also, the Evolutionary Algorithm benefits from integer number order, when its mutation operators are well designed. In the comparison of link selection schemes the ABS scheme came out as a winner. Again, it can be explained by the compactness of the encoding. Another reason may be the fact that each link modification operator is explicitly forced to select a target link. The REL scheme came out as the worst, however, we suppose that when a kind of recurrent tree evaluation operators will be used (see R program-symbol in Gruau (1994)) it may bring results. In the future, it may be also inter-
esting to experiment with a combination of ABS and REL approaches.

The combination of an indirect developmental encoding and proper optimization method is assumed to bring compact genomes able to describe large-scale, modular neural networks. Further experiments with our modifications of Cellular Encoding should be aimed at this direction.

ACKNOWLEDGMENTS

This research is partially supported by the grant Automated Knowledge Extraction (KJB201210701) of the Grant Agency of the Academy of Sciences of the Czech Republic and the research program "Transdisciplinary Research in the Area of Biomedical Engineering II" (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic and by the CTU IGS under grant CTU0812113.

REFERENCES


NAIVE BAYESIAN AND K-NEAREST NEIGHBOUR TO CATEGORIZE ARABIC TEXT DATA

Wa’el Musa Hadi  
CIS Dept  
Arab Academy for Banking and Financial Sciences, Jordan  
whadi81@student.aabfs.org

Fadi Thabout  
MIS Dept  
Philadelphia University, Amman, Jordan  
ffayez@philadelphia.edu.jo

Samer ALHawari  
MIS Dept  
Applied Science Private University, Amman, Jordan  
Samer.alhawari@yahoo.com

Jafar Ababneh  
CIS Dept  
Arab Academy for Banking and Financial Sciences, Jordan  
Jafar_ababneh@yahoo.com

KEYWORDS  
Text Categorization, Naïve Bayesian, Arabic Text Data, K-Nearest Neighbour

ABSTRACT

Text classification is a supervised learning technique that uses labelled training data to derive a classification system (classifier) and then automatically classifies unlabelled text data using the derived classifier. This paper investigates Naïve Bayesian method (NB) and K-Nearest Neighbour algorithm (KNN) on different Arabic data sets. The bases of our comparison are the most popular text evaluation measures. The experimental results against different Arabic text categorisation data sets reveal that NB algorithm outperforms the KNN based on Cosine Coefficient approach with regards to all measures.

INTRODUCTION

With the rapid growth of online information, text categorization (TC) has become one of the key techniques for handling and organising text data collections. TC techniques are used to classify news stories, interesting information on the World Wide Web, and to guide user's search through hypertext. Since building text classifiers manually is difficult and time-consuming, it is advantageous to learn classifiers from examples. The goal of TC task is to assign class labels to unlabelled text documents from a fixed number of known categories. Each document can be in multiple, exactly one, or no category at all. In this paper we focus on just a single label assignment.

Many TC approaches from data mining and machine learning exist such as: decision trees (Quinlan 1993), Support Vector Machine (SVM) (Joachims 1998), rule induction (Moulindier et al. 1996), and Neural Network (Wiener et al. 1995). The goal of this paper is to present and compare results obtained against Arabic text collections using K-nearest Neighbour (KNN) algorithm and Naïve Bayesian (NB) algorithm. The bases of our comparison of the KNN and NB are the most popular text evaluation measures (F1, Recall, and Precision) (Van Rijsbergen 1979). In other words, we want to determine the categorizer that produces the best results.

The organisation of this paper is as follows, related works are discussed in Section 2. TC problem is described in Section 3. In Section 4, experiment results are explained, and finally conclusions and future works are given in Section 5.

RELATED WORKS

Since TC stands at the cross junction to modern information retrieval and machine learning, several research papers have focused on it but each of which has concentrated on one or more issues related to such task. There are few previous works on Arabic TC. For instance, (Khreisat 2006) compared between Manhattan distance and Dice measures using N-gram frequency statistical technique against Arabic data sets collected from several online Arabic newspaper websites. The results showed that N-gram using Dice measure outperformed Manhattan distance.

The author's of (Sawaf et al. 2001) presented results using statistical methods such as maximum entropy to cluster Arabic news articles; the results derived by these methods were promising without morphological analysis. In (El Kourdi et al. 2004), NB algorithm was
applied to classify Arabic web data, the results showed that the average accuracy was 68.78%. (El-Halees 2007) used Maximum Entropy for TC on Arabic data sets, the results revealed that the average F-measure increased from 68.13% to 80.41% using preprocessing techniques (normalization, stopwords removal, and stemming). Furthermore, the algorithm developed by (El-Halees 2007) outperformed other presented text classification algorithms, i.e. (El Kourdi et al. 2004), (El-Halees 2006), (Sawaf et al. 2001), and Sakhr's Categorizer (Sakhr 2004), with regards to F1-measure results.

Finally, (Mesleh 2007) used three classification algorithms, namely SVM, K-NN and NB, to classify 1445 texts taken from online Arabic newspaper archives. The compiled texts were classified into nine classes: computer, economics, education, engineering, law, medicine, politics, religion and sports. Chi square statistics was used for feature selection. Mesleh discussed that "Compared to other classification methods, our system shows a high classification effectiveness for Arabic data set in terms of F1-measure (F=88.11)".

**TEXT CATEGORIZATION PROBLEM**

TC is the task in which texts are classified into one of predefined categories based on their contents. If the texts are newspaper articles, categories could be, for example, economics, politics, sports, and so on. This task has various applications such as automatic email classification and web-page categorization. These applications are becoming increasingly important in today’s information-oriented society.

TC problem can be defined according to (Sebastiani 1999) as follows: The documents divided in two datasets, for training and testing. Let training data set = \{d_1,d_2,...,d_g\}, where g documents are used as examples for the classifier, and must contain sufficient number of positive examples for all the categories involved. The testing data set = \{d_{g+1},d_{g+2},...,d_n\} used to test the classifier effectiveness. The matrix shown in Table 1 represents data splitting into training and testing. A document d_k is considered a positive example to C_y if C_{ky} =1 and a negative example if C_{ky} =0.

Generally, TC task goes through three main steps: Data pre-processing, text classification and evaluation. Data preprocessing phase is to make the text documents suitable to train the classifier. Then, the text classifier is constructed and tuned using a text learning approach against from the training data set.

Finally, the text classifier gets evaluated by some evaluation measures i.e., recall, precision, etc. The next three subsections are devoted to discuss the main phases of the TC problem related to the data we utilised in this paper.

**Data Pre-Processing on Arabic Data**

The data used in our experiments are collected from online Arabic newspapers including Al-Jazeera, Al-Nahar, Al-hayat, Al-Ahram, and Al-Dostor. The dataset consists of 606 Arabic documents of different lengths that belongs to 6 categories, the categories are (Economy "اقتصاد", Health "صحة", Politico "سياسة", Art "فن", Agriculture "زراعة", Science "علوم"). Table 2 represents the number of documents for each category.

**Table 1. Representation of TC problem**

<table>
<thead>
<tr>
<th>Category</th>
<th>Training data set</th>
<th>Test data set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d_1 ... d_g</td>
<td>d_{g+1} ... d_n</td>
</tr>
<tr>
<td>C_1</td>
<td>C_{11} ... C_{1g}</td>
<td>C_{1(g+1)} ... C_{1n}</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>C_n</td>
<td>C_{nm} ... C_{mg}</td>
<td>C_{m(g+1)} ... C_{mn}</td>
</tr>
</tbody>
</table>

**Table 2. Number of Documents per Category**

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>100</td>
</tr>
<tr>
<td>Art</td>
<td>90</td>
</tr>
<tr>
<td>Economy</td>
<td>100</td>
</tr>
<tr>
<td>Health</td>
<td>116</td>
</tr>
<tr>
<td>Politics</td>
<td>100</td>
</tr>
<tr>
<td>Science</td>
<td>100</td>
</tr>
</tbody>
</table>

Arabic text is different than English one since Arabic language is highly inflectional and derivational language which makes monophonal analysis a complex task. Also, in Arabic script, some of the vowels are represented by diacritics which usually left out in the text and it does use capitalisation for proper nouns that creates ambiguity in the text (Hamo et al. 2002). In the Arabic data set we use, each document file was saved in a separate file within the corresponding category’s directory.

Moreover, we represented the Arabic data set to a form that is suitable for the classification algorithm. In this phase, we have followed (Benkhalifa et al. 2001) (Guo et al. 2004) and (El Kourdi et al. 2004) data format and processed the Arabic documents according to the following steps:

1. Each article in the Arabic data set is processed to remove the digits and punctuation marks.
2. We have followed (Samir et al. 2005) in the normalization of some Arabic
letters such as the normalization of (hamza (I) or (1)) in all its forms to (alef (1)).
3. All the non Arabic texts were filtered.
4. Arabic function words were removed.
The Arabic function words (stop words) are the words that are not useful
in IR systems e.g. The Arabic prefixes, pronouns, and prepositions.

**Approaches to Text Categorization**

This section covers two existing approaches to text categorization: K-NN, and NB. K-NN is
applicable to any documents encoding strategy once the similarity between two documents
encoded in such strategy is defined. Although NB is popular approach TC, these approaches
require the representation of documents into numerical vectors which leads to the two main
problems: 1) Huge dimensionality, and 2) sparse distribution. The next two subsections describe
the general nature, process for classifier training and document classification, advantages and
disadvantages, of both learning methods that we consider.

**K-nearest Neighbour**

KNN (Yang 1999) is a statistical classification approach, which has been intensively studied in
pattern recognition over four decades. KNN has been successfully applied to TC problem, i.e.
(Yang and Liu 1999; Yang 1999), and showed promising results if compared with other
statistical approaches such as Bayesian based Network (Tzeras and Hartman 1993).

The KNN algorithm is quite simple: Given training and test documents, the algorithm finds
the k-nearest neighbours among the training documents, and uses the categories of the k-
neighbours to weight the category of the test document. The similarity scores of each
neighbour document to the test document are used as a weight of the categories of the
neighbour document. If several of the k-nearest-neighbours share a category, then the pre-
neighbour weights of that category are added together, and the resulting weighted sum is used
as the likelihood score of that category with respect to the test document. By sorting the
scores of the candidates’ categories, a ranked list is obtained for the test document.

**Naïve Bayesian**

The NB is a simple probabilistic classifier based on applying Baye’s theorem, and its
powerful, easy and language independent method.

When the NB classifier is applied on the TC problem we use equation 1.

\[
p(\text{class} \mid \text{document}) = \frac{p(\text{class}).p(\text{document} \mid \text{class})}{p(\text{document})}
\]

(1)

Where:

\(P(\text{class}(\text{document}))\): It’s the probability of
class given a document, or the probability that a
given document D belongs to a given class C.

\(P(\text{document})\): The probability of a document, we
can notice that p(document ) is a Constance
divider to every calculation, so we can ignore it.

\(P(\text{class})\): The probability of a class (or category),
we can compute it from the number of
documents in the category divided by documents
number in all categories.

\(P(\text{document}(\text{class}))\) represents the probability of
document given class, and documents can be modelled as sets of
words, thus the p(document(class)) can be written like:

\[
p(\text{document} \mid \text{class}) = \prod_i p(\text{word}_i \mid \text{class})
\]

(2)

So :

\[
p(\text{class} \mid \text{document}) = p(\text{class})\prod_i p(\text{word}_i \mid \text{class})
\]

(3)

Where :

\(P(\text{word}_i \mid \text{class})\): The probability that the i-th word
of a given document occurs in a document from
class C, and this can be computed as follows:

\[
P(\text{word}_i \mid \text{class}) = \frac{T_{ci} + \lambda V}{(N_c + \lambda V)}
\]

(4)

Where

\(T_{ci}\): The number of times the word occurs in that
category C

\(N_c\): The number of words in category C

\(V\): The size of the vocabulary table

\(\lambda\): The positive constant, usually 1, or 0.5 to
avoid zero probability.

**EXPERIMENT RESULTS**

We used three evaluation measures (Recall, Precision, and F1) as the bases of our
comparison, where F1 is computed based on the following equation:

\[
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}}
\]

(5)

Precision and recall are widely used
evaluation measures in IR and ML, where
according to Table 3,

\[
\text{Precision} = \frac{a}{(a + b)}
\]

(6)
Recall = \frac{a}{(a + c)} \quad (7)

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Relevant</th>
<th>Irrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents Retrieved</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Documents not Retrieved</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

Table 3. Documents possible sets based on a query in IR

Table 4 gives the F1, Recall, and Precision results generated by the two categorizers (NB, and KNN based on Cosine coefficient) against six Arabic data sets where in each data set we consider 70% of documents arbitrary for training, and 30% for testing, and K parameter in the KNN algorithm was set to 11.

After analysing Table 4, we found that the NB categorizer outperformed KNN on four data sets, and KNN won on a single dataset with regards to F1 results. Precision results obtain that the NB outperformed KNN on two data sets, and tied on 3 data sets and KNN won on single data set, but on average the NB produced 94.31 and KNN 91.37. Also Recall results obtain that the NB outperformed KNN on 4 data sets, and KNN won on single data set. Figure 1, figure 2, and figure 3 depict the F1, Precision, and Recall results on Arabic data sets using NB and KNN, respectively.

Table 4. Results F1, Recall, and Precision of Arabic text categorization

<table>
<thead>
<tr>
<th>Category Name</th>
<th>KNN</th>
<th></th>
<th>NB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>F1</td>
<td>Recall</td>
</tr>
<tr>
<td>Agriculture</td>
<td>100</td>
<td>90.91</td>
<td>95.24</td>
<td>100</td>
</tr>
<tr>
<td>Art</td>
<td>77.78</td>
<td>84.00</td>
<td>80.77</td>
<td>85.19</td>
</tr>
<tr>
<td>Economy</td>
<td>90.00</td>
<td>90.00</td>
<td>90.00</td>
<td>100</td>
</tr>
<tr>
<td>Health</td>
<td>88.57</td>
<td>100</td>
<td>93.94</td>
<td>97.14</td>
</tr>
<tr>
<td>Politics</td>
<td>90.00</td>
<td>100</td>
<td>94.74</td>
<td>96.67</td>
</tr>
<tr>
<td>Science</td>
<td>100</td>
<td>83.33</td>
<td>90.91</td>
<td>83.33</td>
</tr>
<tr>
<td>Average</td>
<td>91.06</td>
<td>91.37</td>
<td>90.93</td>
<td>93.72</td>
</tr>
</tbody>
</table>
CONCLUSIONS AND FUTURE WORKS

In this paper we discussed the problem of automatically classifying Arabic text documents. We used the NB algorithm which is based on probabilistic framework and KNN algorithm to handle our classification problem. The average of three measures obtained against six Arabic data sets indicated that the NB algorithm dominant KNN algorithm. In near future, we intend to propose a new multi-label classification approach based on association rule for the text categorization problem.

REFERENCES


200
ENERGY EFFICIENT PSO-BASED ALGORITHM FOR OPTIMIZING AUTONOMOUS WIRELESS SENSOR NETWORK

Mohaned Al-Obaidy  
Faculty of Computing Awards  
Mobile Computing Division  
Gulf College-OMAN  
Email: mohaned@gulfcollegeoman.com

Aladdin Ayesh  
School of Computing,  
Computer Engineering Division  
De Montfort University-UK  
Email: Ayesh@dmu.ac.uk

KEYWORDS
Sensor Networks, Particle Swarm Optimization, Mobile Swarms

ABSTRACT

Wireless Sensor Networks (WSNs) must be able to operate under very dynamic conditions and, in most cases, in an unattended mode leading to an autonomous network. This scenario becomes more complex when we consider that there will be probably sensor networks that comprise tens of thousands sensor nodes. The design of algorithms for WSNs must consider the hardware limitations of sensor nodes, the physical environment where the nodes will operate, and the application requirements. In this paper we propose to design an algorithm which provides a robust and energy-efficient communication mechanism for large scale mobile wireless sensors network.

INTRODUCTION

The integration of sensing, signal processing, and data communication functions allows a WSN to create a powerful platform for processing data collected from the environment. The algorithms and protocols for this kind of network must be able to enable network operation during its initialization and during both normal and exception situations. While the traffic bandwidth requirement is not the main WSN networking issue, the reliability is strongly expected to be fulfilled [I. Akyildiz(August 2002)]. Any WSN is deeply involved in and related to the monitored environment, and any change occurring to the surroundings will significantly influence its performance; nevertheless, the network must be able to tolerate and ‘survive’ any change by implementing proper reactions and adaptation mechanisms sustaining communications for both sensed data and commands [Bandyopadhyay and Coyle(2003)]. In order to comply with the self-working paradigm, a WSN should implement a set of viable management guidelines, including: [Carlos M.S. Figueiredo and Loureiro(2006)]

- self-configuration, i.e., the ability to automatically and autonomously set relevant parameters to operate according to some given specifications;
- self-organization, i.e., ability to detect the presence of the other network devices and to modify the working behavior, accordingly;
- self-adaptation, i.e., ability to automatically interpret feedback information and to adjust optimal settings to well operate in the environment; and
- self-healing, i.e., ability to detect the failures in the devices and links, by providing autonomous reactions to restore operating conditions without human intervention.

In this work we propose to design an algorithm for a large scale mobile sensors network. This algorithm should provide a robust and energy-efficient communication mechanism which enables the swarms of sensors to move while keeping optimum distances between the sensor nodes.

PRELIMINARY

WSN Basics

A wireless sensor network (WSN) consists of sensor nodes connected among themselves by a wireless medium to perform distributed sensing tasks. This type of networks are expected to be used in different applications such as environmental and health monitoring, surveillance, and security [Laurent Chalard and Zoy(June 8, 2007)]. Sensor networks are a sensing, computing and communication infrastructure that allows us to instrument, observe, and respond to phenomena in the natural environment, and in our physical and cyber infrastructure. The sensors themselves can range from small passive microsensors (e.g., “smart dust”) to larger scale, controllable weather-sensing platforms. Their computation and communication infrastructure will be radically different from that found in today’s Internet-based systems, reflecting the device and application driven nature of these systems. An important aspect of WSNs comes from having many sensors generating sensing
data for the same set of events. To obey the self-working paradigm, WSN protocols should be designed with strong attention to both device coordination and redundancy exploitation issues, both of which might have to cope with the network member resource heterogeneity [Heylighen and Gershenson(May/June 2003)]. A vision to reach this autonomy is through the concept of self-organization, which is defined in [Carlos M.S. Figueiredo and Loureiro(2006)] as "the spontaneous creation of a globally coherent pattern out of local interactions." Local interactions will be probably based on local rules to achieve a global goal. Note that the local rules assigned to each sensor may be different depending on its hardware characteristics, node location, traffic pattern, security, and other attributes associated with the application. The ultimate goal of these local rules is to design a self-organizing WSN.

**Problem Description**

Deployment of mobile swarms can enhance the sensor network in many ways. Firstly, the swarm nodes have much higher hardware capabilities than the sensor nodes. They can provide detailed information of the intended area (e.g. the hot spot). Secondly, the wireless radios of the swarm nodes usually have much longer range and higher channel bandwidth, which can support high quality and delay sensitive multimedia streams. Thirdly, the swarms are mobile [Gerla and Xu(December, 2003)]. They can be easily directed to the hot spots. A limited number of mobile swarms can easily cover a large scale sensor network. Figure 1 illustrates an overview of the proposed sensor network architecture with enhancement of mobile swarms. As we can see, the sensor network can be deployed to cover a very large field due to the low cost of sensor nodes. Sensors nodes perform the basic simple functionalities such as detecting intruders and monitoring the environment changes.

![Figure 1: High density sensors network enhanced by mobile Swarms](image)

**Swarm Intelligence-Possible Solution**

Swarm Intelligence indicates a recent computational and behavioral metaphor for solving distributed problems that originally took its inspiration from the biological examples provided by social insects (ants, termites, bees, wasps) and by swarming, flocking, herding behaviors in vertebrates. It is an attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insects and other animal societies. The common behaviors in all kinds of swarms are [Bonabeau et al.(1999)Bonabeau, Dorigo, and Theraulaz];

- Control is fully distributed among a number of individuals;
- Communications among the individuals happen in a localized way;
- System-level behaviors appear to transcend the behavioral repertoire of the single individual; and
- The overall response of the system is quite robust and adaptive with respect to changes in the environment.

The idea in our proposed system is that; once the sensor network detects something wrong in a certain area, one or more mobile swarms can then be directed to that area for discovering more detailed information perhaps by streaming video and audio data. In Figure 1, the mobile swarms consist of unmanned aerial vehicles which have wireless radios. One is for talking to sensor nodes and one for talking with other vehicles. Another radio may also be installed for each vehicle to talk to a satellite. The satellite can play as a forwarding node to connect different swarms for exchanging information. If no satellite available, we can then build a mobile backbone network among the different swarms. The network of swarms is then connected to the control center.

**Related Work**

In the self-organization of the WSN, two directions have been paid much attention. The former kind is the coverage-based method [Tian and Georganas(2002), Meguerdichian S. and M.B.(April 2001)], which concerns on ensuring the complete sensing coverage with node number as small as possible. Only when one or more operated nodes happen to fail, does the network organization implement once more. It is actually a static method without considering the dynamic of target state. The latter is the distributed collaborative sensing method [Kumar. S and D(March 2002), Chen Wei-Peng and Sha(July/September 2004), Wensheng and Guohong(2003), H. and Sikdar(2006), Feng Zhao and Reich(March 2002)], which constructs an integrated performance index of tracking accuracy.
and communication cost. By optimizing the performance index online, it achieves a tradeoff between the energy cost and sensing performance. However, it usually requires a cluster head and some cluster members to form a centralized construction. Moreover, such centralized optimization may not be practical because each node has very limited computation ability. Besides this, the priori location information of each node is needed beforehand. Clustering wireless sensor networks has been researched intensively in the last decade because this technique can greatly reduce communication cost of the network nodes because they only need to send data to the nearest cluster-head (CH). However, CH expends more energy than ordinary nodes communicating with the sink point (destination). Thus in LEACH [W. R. Heinzelman and Balakrishnan(2000)], Heinzelman et al. proposed a rotation of CHs in each round of communication. In LEACH, CHs are identified by an election process influenced by randomness. Clustering is an NP-hard problem [P. K. Agarwal(1998)]. For a given network it is always difficult to find an optimal CH placement. In LEACH-C [W. R. Heinzelman(2002)], Heinzelman et al. proposed a simulated annealing approach to form clusters and identify CH positions. This method is a centralised solution assuming that the position of all nodes are known in advance and powerful computer performs the computation and inform all the nodes about their respective cluster-heads. In [J. Tillet et al. (2002)], Tillet et al. proposed a Particle Swarm Optimisation (PSO) approach for the same problem. However, the main aim was to reduce an intra-cluster distance by completely ignoring the distance to the sink. In [J. Ji and Li(2004)], Ji et al. applied Divided Range Particle Swarm Optimisation (DRPSO) to optimise weighted clustering algorithm (WCA)[M. Chatterjee and Turgut(April 2002)] parameters. In contrast to the multiobjective optimisation problem, we are devising our problem as singleobjective optimisation. In this paper we propose an evolutionary computing based clustering method to cluster nodes uniformly distributed on a sensor field. We formulate the clustering method as a minimisation problem. The criterion for clustering is derived based on the communication distance between nodes and the CH, and from the CH to the sink-point. The clustering criterion is taken from our previous work [Obaidy et al.(July 2008)Obaidy, Ayesh, and Sheta].

PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization is an evolutionary computing technique based on principle such as bird flocking. This method was first proposed by Kennedy and Eberhart [Kennedy and R.C.(1995)]. In PSO a set of potential solutions are called particles that are initialised randomly. Each particle will have a fitness value, which will be evaluated by the fitness function to be optimised in each generation. Each particle knows its best position pbest and the best position so far among the entire group of particles gbest. The particle will have velocities, which direct the flying of the particle. In each generation the velocity and the position of the particle will be updated. The velocity and the position update equations are given below as (1) and (2) respectively.

\[ v_{i}^{k+1} = wv_{i}^{k} + c_{1}r_{1} \cdot (p_{best_{i}} - s_{i}^{k}) + c_{2}r_{2} \cdot (g_{best} - s_{i}^{k}) \]  

\[ x_{i}^{k+1} = x_{i}^{k} + v_{i}^{k+1} \]  

The parameters used in equations 1 and 2 are described in Table 1.

In recent times, there has been a number of improvements to the original PSO. We have explored different versions of PSO where the extension to the original algorithm is distinct from each other. Following are the PSO versions which are studied in this paper:

PSO - Time Varying Inertia Weight (TVIW)

PSO-TVIW [Shi and Eberhart(1999)] is the basic PSO algorithm with inertia weight varying with time from 0.9 to 0.4 and the acceleration coefficient is set to 2. The time varying inertia weight is mathematically represented as follows:

\[ w = (weight - 0.4) \cdot \frac{MAXITER - iter}{MAXITER} + 0.4 \]  

Where, MAXITER is the maximum iteration allowed, iter is the current iteration number and weight is a constant set to 0.9.

PSO-Time Varying Acceleration Coefficients (TVAC)

PSO-TVAC [A. Ratnaweera and Watson(2004)], proposed by Ratnaweera et al. uses time varying acceleration coefficient (TVAC). The c₁ varies from 2.5 to 0.5 and the c₂ varies from 0.5 to 2.5. Here the cognitive
component is reduced and social component is increased by changing $c_1$ and $c_2$. The large cognitive component and the small social component in the initial stages of the algorithm helps the particle to wander around the search space. However, the small cognitive component and large social component at the later stages of the algorithm helps the particle to converge to the global optima. TVAC is mathematically represented as follows:

$$C_1 = (C_{1\text{min}} - C_{1\text{max}}) \frac{\text{iter}}{\text{MAXITER}} + C_{1\text{min}}$$  \hspace{1cm} (4)$$

$$C_2 = (C_{2\text{min}} - C_{2\text{max}}) \frac{\text{iter}}{\text{MAXITER}} + C_{2\text{min}}$$  \hspace{1cm} (5)$$

In Equations 4 and 5 $c_{1\text{min}}$ and $c_{2\text{min}}$ are constants set to 0.5, $c_{1\text{max}}$ and $c_{2\text{max}}$ are also constants set to 2.5. Thus, in this algorithm as the iter progresses, $c_1$ varies from 2.5 to 0.5 and $c_2$ varies from 0.5 to 2.5.

Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients (HPSO-TVAC)

In this method [A. Ratnaweera and Watson(2004)] the particle behaviour will not be influenced by the previous velocity term of Equation 1. Due to non-influence of previous velocity, re-initialisation of velocity is used when the velocity stagnates in the search space. Therefore, in this method, a series of particle swarm optimisers are automatically generated inside the main particle swarm optimiser according to the behaviour of the particle in the search space, until the convergence criteria is met. The reinitialisation velocity is set proportional to $V_{\text{max}}$. The pseudocode for reinitialisation velocity is as follows:

$$v_{i}^{k+1} = c_1 \text{rand}_1 \cdot (p_{\text{best}_i} - s_i^k) + c_2 \text{rand}_2 \cdot (g_{\text{best}} - s_i^k)$$

$$\text{if}(v_{i}^{k+1} < 0.5)$$

$$v_{i}^{k+1} = \text{rand}_1() \cdot v$$

$$\text{else}$$

$$v_{i}^{k+1} = -\text{rand}_3() \cdot v$$

$$\text{endif}$$

$$v_{i}^{k+1} = \text{sign}(v_{i}^{k+1}) \cdot \text{min}(|f_{\text{abs}}(v_{i}^{k+1}, v_{\text{max}})|)$$

where $\text{rand}_i()$, $i = 1, 2, 3$ are separately generated uniformly distributed random numbers in the range [0,1] and $v$ is the reinitialisation velocity. The effect of HPSO along with TVAC (hence, HPSO-TVAC) on clustering of sensor networks can be observed through simulations.

Particle Swarm Optimisation with Supervisor-Student Model (PSO-SSM)

In this method Liu et al. [Y. Liu and He(2004)] proposed PSO-SSM to achieve low computational costs. The algorithm introduces a new parameter called momentum factor ($mc$) to update the positions of particles. In this algorithm, they also proposed a different velocity updation mechanism from the conventional PSO algorithms. Here velocity is updated only if each particle’s fitness at the current iteration is not better than that of previous iteration. The velocity serves as a navigator (supervisor) by getting the right direction, while the position (student) gets a right step size along the direction. The velocity and the position are modified using the following equations.

$$v_{i}^{k+1} = v_{i}^{k} + c_1 \text{rand}_1 \cdot (p_{\text{best}_i} - s_i^k) + c_2 \text{rand}_2 \cdot (g_{\text{best}} - s_i^k)$$  \hspace{1cm} (6)$$

$$x_{i}^{k+1} = (1 - mc) \cdot x_i^k + mc \cdot v_{i}^{k+1}$$  \hspace{1cm} (7)$$

ENERGY MODEL FOR OPTIMIZATION

We are studying the impact of the transmission range of sensor nodes and positioning of the sink in minimising the communication energy in a sensor network. The important components of each sensor are the data and control processing unit and the radio for communication. The microprocessor used in the processing unit should be energy efficient with very less energy consumption. The energy dissipation in the radio depends on the different characteristics of the radio. The energy model used in this work is adopted from [W. R. Heinzelman(2002), S. M. Guru and Fernando(2005), Wang and Chandrakasan(2002)] and summarised here. The energy dissipation for transmitting $b$ bits to a distance $d$ is shown in Equation (8).

$$E_{tx}(b,d) = E_{\text{elec}} \times b + E_{\text{amp}} \times b \times d^2$$  \hspace{1cm} (8)$$

The energy dissipation in a node to receive $b$ bits of data is shown in Equation (9).

$$E_{rx}(b) = E_{\text{elec}} \times b$$  \hspace{1cm} (9)$$

Where $E_{\text{elec}}$ is the radio energy dissipation and $E_{\text{amp}}$ is the transmission amplifier energy energy dissipation. Energy consumption of a wireless sensor node transmitting and receiving data from another node at a distance $d$ can be divided into two main components: Energy used to transmit, receive and amplify data and energy used for processing the data, mainly by the microcontroller. Leakage current can be as large as a few mA for the microcontroller, and the effect of leakage current can be neglected for higher frequencies.
and lower supply voltage. Assuming the leakage current as negligible, the total energy loss for the sensor system due to the distance $E_{dd}$ can be calculated according to Figure 2 using the following equation:

$$E_{dd} = \left( \sum_{j=1}^{k} \sum_{i=1}^{n_j} (d_{ij}^2 + \frac{D^2}{n_j}) \right)$$  \hspace{1cm} (10)

For more details about the derivation and proof refer to [S. M. Guru and Fernando(2005)].

Figure 2: Energy Model for distance based Sensor Network

**EXPERIMENTS AND SIMULATION**

Refering to Equation (10), we can conclude that by reducing the distance from a node to the cluster-head and the cluster-head to the sink we can minimise the energy dissipation in a sensor network. In our simulation, we cluster the nodes taking into consideration that each node can transmit or receive data from all the other nodes. Thus, nodes considered in this network do not have transmission range constraint. Sensors are clustered using entirely distance based Equation (10). Here the number of clusters are user-defined hence, nodes are clustered for a given number of clusters 'k'. The fitness function for this method is as follows:

$$\text{Fitness} = \min \left( \sum_{j=1}^{k} \sum_{i=1}^{n_j} (d_{ij}^2 + \frac{D^2}{n_j}) \right)$$  \hspace{1cm} (11)

where,

$$\sum_{j=1}^{k} (n_j + k) = N.$$

$N$ is the number of nodes in a network. For our simulations, we used 100-node networks that are uniformly distributed in a 2-Dimensional problem space [0:100,0:100]. We have studied the impact of sink location on the fitness value of the PSO algorithms. In one set of simulations we considered the sink-point to be located at the center of the network (50,50). In another set of simulations we considered the sink-point to be located remotely at (50,180). For both simulations we used the same set of nodes. The maximum number of generations we were running was 1000. The parameters used in the simulations are tabulated in Table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>MAXITER</td>
<td>1000</td>
</tr>
<tr>
<td>$v_{max}$</td>
<td>100</td>
</tr>
<tr>
<td>$x_{max}$</td>
<td>100</td>
</tr>
<tr>
<td>$v$ range</td>
<td>0-100</td>
</tr>
<tr>
<td>$x$ range</td>
<td>0-100</td>
</tr>
</tbody>
</table>

**RESULTS**

In this work we observed the performance in terms of quality of the average optimum value for 10 trials to the PSO-SSM and PSO-TVIW models which are described earlier. We chose these two methods for the following reasons; the PSO-SSM model is the only model which has the ability to stop particles from moving beyond the boundary of the problem space, that is under the influence of $mc$ parameter in it. The PSO-TVIW model is almost similar to the basic PSO algorithm with just the inertia weight varying with time from 0.9 to 0.4. From the graph shown in Figure 3 we can conclude that PSO-TVIW convergence is slower as compared to the PSO-SSM algorithm. This was due to constant acceleration co-efficients used in this model which affects the rate of convergence.

Figure 3: Convergence for the PSO-SSM and PSO-TVIW Models
CONCLUSIONS AND FUTURE WORK

We explored the results of the performance evaluation of four extensions to the standard Particle Swarm Optimization algorithm in order to reduce the energy consumption in Wireless Sensor Networks. Communication distance is an important factor to be reduced in sensor networks. We have simulated two models; the Supervisor-Student Model (PSO-SSM) and the time varying Inertia Weight (PSO-TVIW) model. In the (PSO-SSM) model the new parameter introduced called the momentum factor $w(t)$ to update the position of particles. Also here the velocity is updated only if each particle’s fitness at the current iteration is not better than that of previous iteration. Hence the computational costs for this algorithm will be decreased. An important modification proposed is to use boundary checking for re-initialization of particle which moves outside the set boundary. We can also conclude that (PSO-TVIW) convergence is slower as compared to other algorithm. As a future work, our program can be upgraded with a slight changes to cover the two other models described in this paper, then a comprehensive comparison could done to analyze the behavior of the particles within each case.

ACKNOWLEDGMENT

The authors would like to thank Professor Alaa Sheta from Balqaa Applied University for his support by providing us a guideline for PSO toolbox software.

REFERENCES


[Meguerdichian S. and M.B.(April 2001)]

[Obaidy et al.(July 2008)] Obaidy, Ayesh, and Sheta


[W. R. Heinzelman and Balakrishnan(2000)]


ARTIFICIAL SOCIETY
TRAFFIC BEHAVIOUR
A HIGH PERFORMANCE FRAMEWORK FOR AGENT BASED PEDESTRIAN DYNAMICS ON GPU HARDWARE

Paul Richmond
Dr Daniela M. Romano
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello
Sheffield, UK
Web: http://www.dcs.shef.ac.uk/~paul
E-mail: paul@dcs.shef.ac.uk

KEYWORDS

ABSTRACT

Pedestrian simulations have recently focused on top-down implementations ignoring more computationally intensive agent based dynamics. This paper presents a framework for agent based modelling on the Graphics Processing Unit (GPU) which demonstrates large scale pedestrian simulation and rendering. GPU hardware offers significant performance for dynamic large crowd simulations, however the process of mapping computational tasks to the GPU is not trivial and expert knowledge is required. An agent based specification technique is therefore presented, which allows the underlying GPU data storage and agent communication to be hidden. The framework allows the use of static maps to set static environment obstacles and a zoning technique is described for route planning. Parallel population feedback routines are also used to implement Level of Detail (LOD) rendering which avoids any costly CPU data read-back.

INTRODUCTION

Agent based dynamics are particularly well suited to replicating pedestrian and general crowd behaviours. The individual style of modelling closely resembles the cognitive process of decision making and allows complex interactions to emerge from specifying a number of simple rules. Whilst earlier work (Reynolds 1999) has focused on the specification of such rules in agent based environments, more recent work uses non agent based approximation techniques (Treuille et al. 2006, Courty and Musse 2005) to avoid direct evaluation of inter-agent interactions. Such techniques are computationally cheaper and often map to Graphics Processing Unit (GPU) hardware offering significant performance improvements. This has become particularly important as urban environment visualisation has scaled in accordance to GPU hardware requiring far larger population sizes at interactive rates.

The trend towards utilising the GPU for dynamics simulation is well documented (Harris et al. 2002). The parallel nature of GPU processing offers a massive speed up opportunity for algorithms, which are written with parallelism in mind. In the case of large scale real-time pedestrian dynamics, where populations can reach tens of thousands, utilisation of the GPU is almost a requirement. Without maintaining positional data of pedestrians in GPU memory, the transfer from system memory quickly becomes a significant bottleneck and limiting factor. GPU based alternatives have the significant advantage of maintaining data where it is required for rendering, eliminating this problem entirely.

Despite the obvious performance advantages of harnessing the power of the GPU, a significant disadvantage is the difficulty of mapping computational tasks to the GPUs architecture. Whilst CPU based programming allows simultaneous read write access to DRAM, the data flow of the GPU is more closely resembled by a stream processor. Traditionally this has been made available to programmers though varying programmable stages (the vertex, geometry and pixel stages). Those that use the GPU for general purpose computation, or General Purpose computation on GPUs (GPGPU), have used these programmable stages within a traditional graphics environment by placing data into textures and using the fragment processor to invoke a quad of parallel processes. More recently the introduction of NVIDIA CUDA and unified programmable stage processors has allowed more direct access to graphics hardware, removing the requirement to deal directly with textures and graphics primitives. Whilst this is advantageous the lack of standardisation between hardware vendors makes traditional graphics GPGPU the most flexible option. Further more expert knowledge of the underlying hardware is required in either case to achieve the best performance results.

This paper presents an agent based framework for pedestrian modelling on the GPU using graphics based GPGPU. Its primary goals are to hide the underlying GPU data storage and agent communication, allowing user focus on agent based pedestrian scripting. It builds upon previous work (Richmond and Romano 2008) allowing C like scripting to be used for agent behaviour with a complementary Object Orientated API for setting up initial conditions and initial agent data. Static maps can be set
allowing agent interaction with obstacles which are also used to demonstrate route planning through the environment. Pedestrian animation and rendering is implemented through a key framed animation technique with a level of detail system which is maintained entirely on the GPU. Whilst the agent based pedestrian dynamics used in this paper are based on simple but well established work (described in the following section), it is advised that the described framework is suitable for inclusion of more advanced cognitive and emotional models (Romano et al. 2005).

AGENT BASED PEDESTRIAN DYNAMICS

Agent Base Modelling (ABM) of dynamic systems gained popularity through Craig Reynolds who, in 1987 demonstrated flocking bird behaviour through an agent based Boids model (Reynolds 1987). Unlike dynamic systems controlled through sets of partial differential equations, ABM examines the low level individual behaviours which when part of a social system result in emergence of non predictable social events. In the case of the Boids model the low level behaviour can be described by three simple rules; separation at close range, attraction to the perceived flock centre and velocity matching of perceived neighbours. This concept is extended in Reynolds later work (Reynolds 1999) to include advanced behaviours such as pursuit, evasion, obstacle avoidance and flow field following for general agent locomotion.

Helbing (Helbing et al 2002) introduced the social forces model as a class of equations which balance various environmental and social influences on pedestrians in both normal and panic situations. Essentially this builds upon previous work (Helbing and Molnar 1995) which mathematically formalises the concept of social forces as a behavioural description of conflict situations. Most important in this model is the description of a repulsive social force acting between pedestrians. In its simplest form, this force acts as a mechanism for collision avoidance giving greater precedence to situations in font of the pedestrian, over situations occurring behind. Additionally longer range attractive environment forces such as attraction to window displays, special attractions or acquaintances are modelled using a force summation which are combined with the social repulsive force. In emergency situations an additional short range body force and sliding friction force is described which avoids compression in high density populations. Whilst simplistic with respect to the dynamical rules, complex natural pedestrian activity can be observed in Helbings (Helbing and Molnar 1995) work. Phenomenon such as lane formation and intersection dynamics are just two of the emergent behaviours empirically observed in real world conditions.

Building upon Helbings (Helbing and Molnar 1995) work, whilst considering large population sizes, Quin et al. (Quin et al. 2003) demonstrated a technique which used a SWARM cluster of 10 processors to simulate 10,000 evacuating pedestrians using the social forces method. Much attention in this work is given to a spatial partitioning technique which distributes pedestrians across the processing node with communication between them handled with a Message Passing Interface (MPI) library. This technique, as also demonstrated more generically by the Flexible Large-scale Agent Modelling Environment (FLAME) (Adra et al 2008), is one which inspires this work. The use of spatial partitions is essential in producing large simulations and scalable algorithms, particularly with respect to pedestrians where there is no need for total communication between all agents.

Coutry and Musse (Coutry and Musse 2004) present a GPU implementation of pedestrian dynamics which is inspired by Helbings social forces model (Helbing and Molnar 1995). Its uses a cellular system which allows agents to move between empty cells in discretely partitioned space. As each cell occupies at most a single agent the social repulsive force is calculated on the fly during the update stage by considering a neighbourhood of surrounding cells. Improving upon this by avoiding the potential of agents occupying the same grid space, Coutry and Musse (Coutry and Musse 2005) propose an implementation of a dynamically created force field texture. This technique uses GPU hardware blending to compute the sum of all forces for each discrete partition. The result is rendered to an off-screen texture, the force field texture, and is then used during the update stage to avoid direct communication between pedestrians is the system. Similarly to this Treuille et al. (Treuille et al. 2006) uses the same technique to construct a density field of pedestrians which is in turn used to determine a more complex discomfort field. The discomfort field is constructed in real time by considering local collision avoidance and global path planning. Performance of this technique is dependant on the resolution of the discomfort grid. With a grid of 60° Treuille (Treuille et al. 2006) reported performance of 10,000 pedestrians 5fps before rendering in-between frames.

As recent pedestrian modelling has tended towards less agent based techniques, it is no surprise that recent high performance agent based simulations are concentrated on more general agent based and swarming applications. Reynolds more recent work on PSCrowd (Reynolds 2006), demonstrated through schools of fish, is analogous to pedestrian behaviour and presents a 2D environment performance of 15’000 low resolution (36 polygon) fish at 60fps. The technique used to implement this is based on the previous work (Reynolds 1999) and uses a spatially partitioned 3D or 2D environment. The high performance speed is achieved by utilising the Playstation3’s cell processor to batch a number of spatial buckets across to the cell processors 8 parallel Synergistic Processing Units (SPU’s). Unlike Reynolds earlier work, the neighbourhood heuristic in this case is based on N-nearest neighbour, with each agent maintaining its own neighbourhood list. As interaction is not based on a defined radius or limited vision, this technique is highly dependant on the size of the neighbourhood lists.
More general agent based work has recently been demonstrated by D’Souza et al. (D’Souza et al. 2007). This work concentrates on implementing a number of 2D agent based models on the GPU with particular attention to performance. Similar to the previously described methods, spatial partitioning is used however each partition is responsible for maintaining only a single agent. Although this likens the system to that of cellular one, agents have continuous values and collisions (multiple agents occupying the same space) are resolved using a multi pass priority system. Agent communication is achieved in the same fashion cellular automaton and performs an expanding search into neighbouring cells. Whilst this technique boasts a performance of up to 2 million simple agents in real time, it is important to understand the limitations of such a technique when applied to a pedestrian system. Firstly fine grained partitioning requires large amounts of space. If a larger partition size is used, space is saved with the trade-off that there will be more costly collisions to resolve. Secondly if agents are well distributed, particularly into local groups across a large area then large amounts of partitioned space are wasted by holding no useful data. Decreasing the partition size again helps to resolve this however as in the first case higher concentrated areas have to then resolve additional collisions.

AGENT BASED MODELLING ON THE GPU

As this work aims to present a useable framework for agent based pedestrian specification, rather than a single simulation example, the mapping of agent data to the GPU is a process which it is important to abstract. The idea behind this is that both agent scripting and the programming interface should be able to get and set agent data variables without explicit knowledge of the underlying GPU memory architecture. This is achieved through a translation function ‘F’ which provides a mapping for both agent update scripts and the getting and setting of initial agent data into a number of 2D stacked 32bit floating point textures (agent space). Similarly to previous work on particle systems (Latta 2004, Kipfer et al. 2004) a 1D list of variables is easily translated into 2D texture space with a 2D position \((i, j)\) in each stacked texture ‘r’ representing an individual set of data. As texture access is read/write only, ‘2r’ textures are required in total with data being stored in up to all four of the red, green, blue and alpha colour channels respectively (figure 1). Within OpenGL the Frame Buffer Object extension allows simultaneous writing to Multiple Render Targets (MRTs), on our implementation hardware (a NVIDIA GeForce 8800 card) up to 8 targets are supported giving a total of 32 agent variables. For a potential non communicating system of agents, simulation is achievable by performing \(N\) parallel operations by rendering a single quad primitive. Assuming the quad primitive is the same dimension as agent space and rendered from an orthogonal perspective, rasterisation will invoke a parallel operation for each of the \((i, j)\) agent positions.

In many of the previous examples partition space is used to either directly hold agent data or hold pointers to data in agent space. Similarly this work splits space into discrete spatial partitions however the partition size is equal to the agent’s communication radius with each partition holding any number of agents. Agent communication can thus be achieved by consulting all agents within an agents own and neighbouring partitions. On the CPU this task is trivial and each agent in a partition can be stored as a linked list. On the GPU this is significantly more difficult as dynamic storage sizes are unsuitable for parallel computation. Instead a dynamic structure is computed before each agent update and adapting techniques used in rigid body particles physics (Harada 2007, Green 2007) a matrix is calculated allowing each agent within each spatial partition to be located in agent space. The first stage of the technique involves sorting the agents in agent space depending on their location. As with particle based work (Kipfer et al 2004, Green 2007) each partition is assigned a unique spatial bin identifier based on their \(x\) and \(y\) position. Agents can easily determine their location identifier and are sorted using cache efficient bitonic sorting (Govindaraju et al. 2005) into a theoretical 1D (stored physically in 2D texture space) list according to this. A vertex scattering technique is then applied, where for each agent a vertex is drawn at the origin with texture coordinates allowing each of the agent sort values to be looked up from agent space. Each sort value is compared to that of the previous agent and if the sort value is the first in the sorted 1D list then a linear search is performed until the last sort value is also found. Assuming an off-screen buffer size with the same dimensions as partition space is used, the start and end positions of each partition identifier can be scattered by offsetting the vertices position with the start and end indices preserved as texture coordinates. Vertices which are processed and are found not to be boundary indices are simply positioned outside of the off-screen buffer and ignored. The end result of this is that each occupied
partition space in the off screen matrix contains the start and end positions of all agents within it. This is then used in the same way as a CPU list to iterate through each agent testing for agents within the communication radius. Whilst not all agents tested will be suitable for communication this improves dramatically on the worst case O(n^2) and unlike nearest neighbour techniques guarantees communication for all agents within the set radius.

**AGENT BEHAVIOUR SCRIPTING**

Use of the agent based pedestrian API is dependant on an appropriately defined agent update script. Agent scripts are defined in a C like environment and must override a function, agentMain, accepting an agent and global variables structure as arguments and returning a single agent structure. As reference to variables in the agent structures are mapped at compile time with the previously described mapping function, agent update scripts are free to reference agent variables directly. With respect to communication, agent scripts use two placeholders FOR EACH_AGENT_A and END FOR EACH to hide the underlying algorithm. Further vision test can be applied to each agent if the desired behaviour requires this. For agent populations with no environmental influences this is all that is required to compute an agent’s new internal variables and hence provide a convincing simulation. The globals argument to the main update function offers an extension to this by allowing global static variables to be set between each update stage. Single Float and 2D data array values can be set using the programming interfaces AgentPopulation class. In addition to this role the class also handles the initial setup, agent script compilation (to valid Cg code), agent initialisation and stepping of the simulation loop.

Using the API to simply get and set global 2D data it is possible to encode environmental forces (Courty and Musse 2005) avoiding the costly computation of each agent wall combination as described in Helbing model (Helbing and Molnar 1995). As the granularity of this environmental force field texture is independent from agent interaction large grained force fields can be used, providing they have sufficient detail to capture the smallest static obstacle. 2D data arrays are stored in GPU memory as textures and as a result up to 4 values can be stored in each array. For the purposes of obstacle avoidance the first two components of the array (red and green) are enough to hold a directional velocity. In the remaining components of the array the examples in this paper store a height map value, used for rendering static geometry at a later stage and a non regular environment space identifier. This identifier has the purpose of zoning the environment to allow longer range path planning where pedestrian’s behaviour is beyond that of a random walk. Figure 2 demonstrates an environment map with (black) static obstacles split into 6 zones. Two additional 2D data sets encode firstly a navigation lookup grid which indicates the next zone to move into from the current zone (vertical) to the desired long range goal zone (horizontal). Secondly an additional set of data is required to store an x and y (goal) point value for each of the zones. These are used as goal points which the agent will move towards. In environments where zones consist of complex obstacles with narrow gateways such as door ways, experimentation has shown that the doorway itself must use a zoned area. This step ensures that pedestrians intelligently pass through gateways avoiding a situation where there desired path is blocked.

![Diagram of zoned environment with corresponding data tables](image)

**Figure 2 – A simple illustrative zoned environment with corresponding data tables**

**PEDESTRIAN BEHAVIOUR**

The pedestrian behaviour in this paper is influenced mainly by Reynolds work (Reynolds 1999) however as many factors draw strong parallels with Helbings social forces model (Helbing and Molnar 1995) the exact rules are somewhat of a hybrid. Formally the following equation (1) is able to describe the force exerted on each pedestrian during the update stage for all examples within this paper.

\[ F_i = R_i + Cr_i + G_i + M_i \]  \hspace{1cm} (1)

The total force exerted on each agent \( F_i \), is the result of a social repulsion force \( R_i \), a close range interaction force \( Cr_i \), a short term goal force \( G_i \) and an environmental force field force \( M_i \). This force is then used as a directional steering force to make some change to the pedestrian internal velocity. This altered velocity is then checked to ensure it does not exceed the pedestrians maximum velocity and if required the velocity is normalised accordingly. The social repulsion force uses the same preference to events in the direct line of sight as in Helbings (Helbing and Molnar 1995) model. The symbol \( \lambda \) is simplified to represent a scalar value indicating the size of angle between the pedestrian i’s line of sight and pedestrian j’s position. Additionally as in Reynolds work (Reynolds 1999) agents
are given a limited vision which filters agents outside their field of view. More formally the following equation (2) describes the force \( R_i \), where the letter \( j \) represents only agents from within the limited vision filter.

\[
R_i = S \sum_j^6 \lambda_j \left[ \frac{l}{|P_i - P_j|^2} \right]
\]  

(2)

The static value \( S \) indicates a scalar value controlling the influence of the social repulsive force. The positions \( P_i \) and \( P_j \) represent the positions of agent \( i \) and \( j \) respectfully and the distance between them represents the directional force vector between the two agents. This force is further scaled by the inverse square of the distance between the two agents. The value \( l \) is used to scale the effect of the inverse distance effect and in most examples has been tweaked depending on the interaction radius between agents. Unlike the social repulsive force, the force \( C_i \) is independent of the direction between agents. Its influence is over a far smaller radius are rarely has effect on pedestrians unless there is a high concentration in which it acts mainly as collision avoidance. The force \( G_i \) acts as an influence towards a specific goal point in the environment. In the case of a random walk the goal position is directly in front of the pedestrian encouraging them to follow their current path. In cases involving longer range goals, this goal value is found as a result of environmental lookups as described in the previous section.

**PEDESTRIAN VISUALISATION**

Pedestrian rendering is achieved through two very different methods. The first of these uses a primitive directional triangle. In this case each triangle primitive is rendered at the origin with multi texture coordinates reflecting an agent position in \((i, j)\) agent space. A vertex shader then looks up agent positional and velocity data and translates and rotates the primate object accordingly. As an entire population represented as primitive objects contains relatively few OpenGL calls in total, the entire population can be stored in a single display list. For more advanced pedestrian representations where individual model sizes become much larger this technique quickly becomes unsuitable. In such cases it is necessary to store only a single model representation in a display list. The single display list is then called for each agent with the multi texture coordinate value set before the display list is called, allowing each instance of the model to be translated by a differing set of agent values.

As pedestrians do not move as static objects it is important to animate them with walking behaviour as they move around the environment. Key-framing provides an ideal animation technique as it is computationally cheap and is easily capable of representing simple human locomotion. Through experimentation it is evident that reasonable walking behaviour can be achieved through interpolation between only two key frames. For improved fidelity multiple key frames can be used however as all draw calls are stored in a single display list it is necessary to store the positional and normal information for each key frame model in the list. Whilst this has a visually improved effect on animation of close pedestrians the overall performance degradation makes interpolation between two key frames the preferred option.

With the previous technique every pedestrian in the population is rendered with the same detail level. A more suitable technique is to therefore apply a LOD rendering system which varies the pedestrian’s fidelity depending on distance to the viewer.

This is achieved through the use of a generalised feedback system available for retrieving data about the agent population without CPU read-back from the graphics card. Parallel reduction is used to reduce values in agent space to singular values for a number of common reduction functions such as minimum, maximum, sum and count. For the purposes of a LOD system it is required that the total number of each detail level is know. An agent variable therefore is used to hold a LOD level and is calculated during the agent update stage. A reduction function for each detail level then uses a filtered count function, counting only the number of occurrences of the specified detail level. After the parallel reduction is complete the agent data must then be sorted according to the LOD levels. This ensures that calling a display list for each detail level, the number of times reported by the feedback step, matches the detail levels to the agent data. Whilst this technique is computationally more expensive due to the secondary sort, it allows a massive reduction in rendering overheads when high resolution models are required. This technique is demonstrated in figure 3 which shows pedestrians coloured by their corresponding LOD. Additionally the same technique can be applied to achieve variance in pedestrian representation. In this case varying pedestrian models each with an associated value are used within the population with the value used as feedback and sorting key. This technique also allows simultaneous LOD rendering as long as each LOD pedestrian representation combination has an associated identifier and display list representation.
PERFORMANCE AND RESULTS

In order to test the simulation and rendering performances a range of pedestrian population sizes have been tested using varying rendering fidelities. The results obtained are based on a single PC with an AMD Athlon 2.51Ghz Dual Core Processor with 3GB of RAM and a GeForce 8800 GT. In all cases pedestrian behaviour incorporates all forces described in equation 1. Figure 4 additionally demonstrate a more complex force map representing the Peace Gardens area of Sheffield city centre. The results of a pedestrian simulation compared to satellite imagery are also show in figure 9. Simple zoning is demonstrated in figure 5 and has an immeasurable performance effect on the simulation compared to the following results presented in this section.

Simulation performance is considered by increasing the population size whilst maintaining a roughly static population density. Figure 6 demonstrates the population density by showing the number of pedestrians considered for communication (lookups) and the number actually inside the pedestrians communication radius (communications). As the population density is constant the communications to lookup ratio remains at roughly 35% in all cases. Figure 7 shows a performance chart which demonstrates the effect of increased population sized on performance. Two pedestrian vision (or interaction) radii are used, the first of 4m is suggested in Helbings work (Helbing and Molnar 1995) whilst the second 32m radius acts to demonstrate the performance in cases of longer range social force planning or higher congestion population densities. In both cases an environment force map is used to simply direct pedestrians away form the outer edges of the environment. From the results it is clear that the simulation performance which also includes the basic direction triangle rendering is suitable for

Figure 3 - 65'000 fully interacting agent based pedestrians rendered by LOD level

Figure 5 - Zoning around a congestion point (Black to White boundaries represent walls)

Figure 4 - A Force Map of Sheffield Peace Gardens
large population sizes. More specifically interactive population sizes of 262'144 pedestrians can be maintained at 13 fps or 65'536 at 42 fps for a 4m pedestrian vision.

![Graph showing pedestrian vision compared to pedestrian lookups and inter-agent communications.](image)

**Figure 6 - Pedestrian vision compared to pedestrian lookups and inter-agent communications**

![Graph showing simulation and rendering performance for advanced pedestrian rendering at various detail levels.](image)

**Figure 8 - Simulation and rendering performance for advanced pedestrian rendering at various detail levels**

In order to consider more advanced rendering polygon models of increasing complexity have been used (detail level 0 represents lowest complexity, detail level 2 the highest). Figure 8 demonstrates that even at relatively high population sizes of 16384 pedestrians, high detail models (level 2) of up to 1000 polygons can be used at 40fps. The more modest pedestrian representation of 400 polygons (level 1) with the same population size achieves a performance of 40 fps and the more intelligent dynamic LOD rendering improves further achieving 50fps. What is noticeable is the cost of using a dynamic LOD for agent population sizes of less than 1024. In fact when agent sizes are this small the overhead of the additional sort step suggests that high resolution models for agents can be used whilst sustaining a performance of almost 200fps.

![Satellite imagery of Sheffield Peace Gardens (Left) and our simulation (Right).](image)

**Figure 9 - Satellite imagery of Sheffield Peace Gardens (Left) and our simulation (Right)**
DISCUSSION AND FUTURE WORK

In summary this paper has presented a high performance agent based pedestrian simulation. Further more the work has been designed with a scripting based interface to allow the incorporation of more advanced agent based behaviours in the future. As pedestrian data and simulation has remained entirely on the GPU rendering of pedestrians has been considered. Using real time feedback a LOD system has been presented which has improved the performance and allowed high detail models of pedestrians to be used with large pedestrian population sizes. Whilst pedestrian behaviour has demonstrated both long range inter pedestrian interactions and path planning using zoned areas the work could benefit from the inclusion of more advanced long range path planning as in (Petré et al 2006). Other future work will concentrate on integration with large scale city modelling and automatic path finding within city environments. Pedestrian representation will also be addressed by considering individualised walking styles through kinematics and improved variance across the population through the use of more varied polygon model templates.

REFERENCES


Latta, L. 2004. “Building a Million Particle System”, In proceedings of Game Developers Conference, San Francisco, CA


Reynolds, C. W. 1999. “Steering Behaviors For Autonomous Characters”. In the proceedings of Game Developers Conference. San Jose, California. Miller Freeman Game Group, San Francisco, California. pages 763-782.


MODELING PEDESTRIANS’ MOVEMENT REGARDING ROAD-CROSSING COMPLIANCE
Carlos O. Prettó; Helena B. B. Cybis and Paula Ariotti
Production Engineering Department
Federal University of Rio Grande do Sul - UFRGS
Av. Osvaldo Aranha, 99 – 5º andar, Porto Alegre 90.040-020
+55 51 33083909
Brazil
cprettó@producao.ufrgs.br
helenabc@producao.ufrgs.br
paula@producao.ufrgs.br

KEYWORDS
Pedestrian, Crosswalk, Agent simulation.

ABSTRACT
In this paper, we address the problem of modeling the pedestrian road-crossing task using multi-agent techniques. We propose to model patterns of behavior at road crossing, regarding the pedestrian’s decision-making process about where and when to start crossing the road. First, we classified the pedestrians according to their behavior while crossing the road and, then, we built the pedestrian movement model regarding that classification of behavior.

INTRODUCTION
Pedestrians are the most vulnerable users of the road system. The provision of adequate and safe circulation systems can reduce traffic accidents, especially those involving human beings. It can also reduce their discomfort due to the daily experience of risky situations.

An appropriate infrastructure for pedestrians and cyclists encourages the use of these modalities as an alternative to motorized transport. Improved conditions for getting around qualify the trips in which traveling on foot or bicycle are complementary to motorized trips. The design of projects concerned with pedestrian circulation should take into account the users’ preferences and perceptions in order to improve the effectiveness of these structures (Sisiopiku and Akin 2003). A safe and pleasant road system basically involves five elements (Sarkar 2003): the physical separation of pedestrians and the motorized traffic; the control of pedestrians and vehicles flow; visibility; adequate communication through signaling, and help for pedestrians with special needs.

An appropriate design of pedestrian structures based on these elements may encourage walking without compromising the users’ safety and convenience. Pedestrian safety and comfort may be affected by operational changes, as for example, in traffic signal timings. Pedestrian crossings are essential elements of the transportation system. Properly designed and located crossings perform two important functions: to create expectation among the drivers as to where the pedestrians might cross the road, and to encourage pedestrians to use the structures appropriately.

The emulation of pedestrians in traffic environment is a complex problem (Blue et al. 1997). Some authors use artificial life and cellular automata approach to model the rich complexity of the real movement of pedestrians, the interaction with each other, path planning, sense and the avoiding of obstacle and physical constraints (Hamagami and Hirata 2003, Ronald and Sterling 2005). There are authors who combine the use of agent-based paradigm and cellular automata modeling techniques. The majority of lines of research on pedestrian modeling rely on crowd simulation and pedestrian movement and interaction in urban environment (Shao and Terzopoulos 2007, Jian et al. 2005). Another line of interest is pedestrian behavior on road-crossing activity (Ashida et al. 2001, Zhang and Duan 2007, Liu 2000). We aim to combine these two lines of research into a new model of pedestrian agent, incorporating the concept of crossing compliance.

Pedestrians are classified according to their behavior while crossing the road. Appropriate crossing behavior is defined as crossing compliance (Rouphail 1984). Crossing compliance can be classified according to two types: spatial, related to the location where pedestrians cross the road and temporal, related to the time when the pedestrians decide to cross the road.

The emulation of a more realistic pedestrian behavior pattern provides the basis for the identification of the influence of operational attributes, such as signal timing and project characteristics and the dynamics of the pedestrians’ road-crossing process as a whole.

The road-crossing compliance classification method will be presented in the next section, followed by the first part of the pedestrian model based on agent simulation techniques, and finally by the preliminary results and conclusions.

CROSSING COMPLIANCE
Pedestrians have several choices before crossing a road. The decision about where and when to cross is influenced by the conditions of the road environment as a whole. From the point of view of traffic engineering, the main factors influencing operational and user safety perspectives involve the pedestrians’ preferred crossing location; the conditions under which they decide to cross the road, and the characteristics of traffic flow.

The pedestrian’s choice of a place to cross may be related to three main factors: the pedestrian’s wish to walk shorter
distances; the wish to cross the road as quickly as possible, and the need to reduce exposure to risk. Literature suggests that the pedestrian’s behavior when crossing a road characterizes crossing compliance (Rouphail 1984). Crossing compliance measures the adequate use of the pedestrian facility. In the case of signalized crossings, two compliance concepts can be defined: spatial and temporal. Literature suggests a classification of pedestrians related to spatial crossing compliance, classifying them according to the concept of temporal crossing compliance (Rouphail 1984). Based on these considerations, the conceptual model developed in this study combines both approaches and proposes a structure to explain pedestrian crossing behavior, according to the place and time chosen.

**Spatial Crossing Compliance**

The concept of spatial crossing compliance is based on the hypothesis that each crossing has an area of influence which attracts the pedestrians who cross a given road. An area of influence of the crossing includes half the distance between two consecutive crossings, considering an imaginary line from the central point of each crossing facility, multiplied by the road width. The proper crossing area covers the part of the road located within three meters to both sides of the crossing structure, multiplied by road width. As to spatial compliance, the pedestrians can be classified according to two types:

- prudent: those who cross at the appropriate location, i.e., in the crossing area, and;
- imprudent: those who cross outside the crossing area.

**Temporal Crossing Compliance**

Temporal crossing compliance (TCC) is a measure of the percentage of pedestrians that obey traffic signal orientation. This measure provides information about pedestrians’ patience regarding the duration of traffic light. Thus, regarding temporal crossing compliance, pedestrians can be classified into two categories (Liu 2000):

- obedient: pedestrians who cross during pedestrians’ green light, and;
- opportunist: pedestrians who cross the road during red light, if they find an acceptable gap in the traffic flow.

**Crossing Compliance Classification**

The concepts of crossing compliance presented in the previous sections allow the establishment of a conceptual model to analyze pedestrian behavior at traffic signal. Figure 1 shows the schematic description of the model involving the four groups of pedestrians.

![Figure 1: Structure of the conceptual model of analysis of pedestrian behavior](image)

Pedestrians who comply with the crossing are represented by the prudent and the obedient. Impudent and opportunist pedestrians constitute the group of users who are potentially at risk. However, it should be considered that crossing compliance is directly related to a series of factors: local characteristics, traffic composition, vehicle flow regime and the nature of the conflicts that occur at the crossing.

**PROPOSED MODEL**

The interaction of real pedestrians has influence on their walking behavior. Due to their characteristics, pedestrians can move around with more flexibility on a road environment than if using any other means of transport. Pedestrians have more flexibility to overcome obstacles. They need to avoid or overtake other pedestrians to maintain their speed, change directions, stop and wait. Generally speaking, they tend to keep a minimum distance from another pedestrian or obstacle. The model was divided in three levels of simulation: macro, meso and micro. The macro level regards the initial and final location of the pedestrians, considering only the Euclidean distance between them. The meso level is responsible for the route generation. The micro simulation layer is concerned with space negotiation between pedestrians and the task of avoiding obstacles between them. The three simulation levels are described below.

**Macro Simulation Level**

The macro simulation level considers only the initial and final point of each pedestrian in simulation (Figure 2). Each pedestrian is created with an origin and a destination. Origins are defined by a point with coordinates X and Y. Destinations may be defined by points or areas. This information (initial and final points) is the input of the meso level.

![Figure 2: Macro Simulation](image)
Meso Simulation Level

The meso simulation level is responsible for identifying the best route to reach the destination. This simulation level is defined by a layer of nodes that are connected to the neighbor nodes by two-way links. Each node of the graph represents a reference point of the simulation space. The graph layer contains only a partial number of the fine grid points (Figure 3).

Figure 3: Micro Level Grid and Meso Level Graph Layer

Using the graph layer, each pedestrian solves a route problem from the initial point to the final point supported by A* (a-star) algorithm.

A* algorithm

A* is a best-first, graph search algorithm that finds the least-cost path from a given initial node to one goal node. To solve the pedestrians’ route problem, is used a distance heuristic function to determine the order in which the search visits nodes in the graph. The algorithm was first described in 1968 by Peter Hart, Nils Nilsson, and Bertram Raphael.

Micro Simulation

The micro simulation level considers the step-by-step pedestrians negotiation. Each pedestrian perceives the world and avoids the collision with obstacles and other pedestrians. The pixel collision is used to identify when a pedestrian will collide. The pedestrian decides the best next step considering a combination of alternatives, defined by two parameters: distance module and an angle step. For instance, with an angle step of 15 degrees and a distance module of 10 units, the pedestrian will evaluate 24 possible positions (10 units distant from the actual position) for the next step, regarding the meso simulation route, final point and other pedestrians. In Figure 4, circle A1 represents an agent and circle A2 represents another agent. Using pixel collision detection, it is possible to identify collision and also to perceive the presence of another pedestrian around before the collision.

Figure 4: Pixel Collision

Space and Time

Space is represented by a fine grid (micro level) and a graph layer (meso level). The pedestrian is represented by a circle with a defined radius (Figure 4). Time in the model is represented by a number which increases with each main loop interaction. Each pedestrian has an internal count that represents a period of time to wait until the next step. For each interaction of the main loop (simulation loop), the software runs a list of pedestrians, and if the period time of waiting has been completed, the pedestrian will take the next step.

Pedestrian’s Model

Pedestrians are generated with an implicit profile. Regarding spatial compliance, pedestrians may be prudent or imprudent. When a prudent pedestrian generates his graph, only the safety area is considered (Figure 5). By definition, the “Lane” area is not part of the prudent pedestrian path. Therefore, this zone has no graph nodes. Although pedestrian entrance in the lane is not physically forbidden, it will only happen for a few steps. This mechanism is better understood looking at the pedestrian’s step decision process.

Figure 5: Safety Zone

Step Decision

The step decision process unifies the meso and micro simulation levels. The pedestrian tries to walk through the route generated by A* algorithm (meso level). At the same time, they need to avoid other pedestrians and obstacles (micro level). The pedestrian’s next step is defined by a sequence of decisions:

Decision 1 - Route and next point to reach:

1. Pedestrians try to follow the minimum cost route, subject to the micro level simulation constraints.
2. If a point out of the route is reached: the pedestrian evaluates his distance from neighbor nodes within the meso simulation level.
3. If the closest node belongs to the minimal cost route: he keeps following the route.
4. If the closest node does not belong to the minimum cost route: he generates a new route and tries to walk through it.

Decision 2 – Step:

1. The pedestrian chooses the next immediate goal evaluating a set of possible X and Y coordinates
considering a distance module and an angle step. Pixel collision is used to identify other pedestrians and obstacles.

2. The best position is evaluated based on the Euclidian distance to the immediate goal decided on the “Decision 1” process, and the presence of other pedestrians.

3. The pedestrian moves to the next position.

Figure 6 describes the “Decision 1” process.

![Decision 1 Process Diagram]

Figure 6: “Decision 1” process

The path generated by the pedestrian can be visualized at Figure 7. The pedestrian starts at point “A”. The red line is the first route generated based on start point “A” and final point “B”. The green smooth line generated by the micro simulation process is the real path followed by the pedestrian. At point “X”, the pedestrian finds an obstacle that forces him to diverge from the original route. After this change, the pedestrian reaches a node out of the original route, a point close enough to be considered a new start point. Then, the pedestrian generates a new route (blue path) and follows it to the final point (B).

![Pedestrian Path Diagram]

Figure 7: Pedestrian path

PRELIMINARY RESULTS

Two scenarios were created to evaluate the model results. The first one simulates two prudent pedestrians on a crosswalk task. Each one tries to reach the other side of the zebra. In this simulation, one of the pedestrians changes his original path to avoid the collision with the other pedestrian. It is demonstrated in Figure 8, where Ped 2’s series needs more time to reach its goal (red curve).

The second scenario simulates six prudent pedestrians trying to crosswalk the zebra, three of them starting on each side. Figure 9 shows a distance/time chart, where the “y” axis represents the remaining distances for pedestrian to reach their goal, and the “x” axis represents the simulation time. The average speed of pedestrians in the simulations is 1 m/s. Figure 10 shows the difference between the minimum travel time for pedestrians to cross the zebra and the journey pedestrians faced considering the delay caused by the interaction with other pedestrians. The graph represents the effect of the increase in the number of pedestrians (2,4, and 6) in the travel time.

![Distance-Time Chart]

Figure 8: 1 x 1 Pedestrians in a Collision Route.

![Distance-Time Chart]

Figure 9: 3 x 3 Pedestrians in a Collision Route.

![Travel Time vs. Minimum Travel Time]

Figure 10: Real Travel Time versus Minimum Travel Time
CONCLUSIONS

This study presented a conceptual model devised to improve the understanding of pedestrians’ behavior pattern. The model was constructed by aggregating different approaches of pedestrians’ behavior. The modeling approach presented here provides a simple and reliable way of representing pedestrians’ behavior when crossing a road. The agent-based simulation approach provided the basis for representing the pedestrians according to their behaviors.

Based on the preliminary results, some conclusions follow:

- The proposed classification of the simulation into macro, meso and micro levels provided an easier way to solve the route problem and pedestrians’ space negotiation process;
- The graph layer was able to solve the problem of pedestrians’ route;
- Spatial and Temporal Compliance are useful concepts for understanding pedestrians’ behavior and providing a more realistic simulation of the crossing-road process.

In the future, to achieve a more realistic overall behavior for the simulation, some aspects should be considered, such as:

- A heuristic algorithm for path planning considering a larger pedestrian’s neighborhood;
- The inclusion of other variables to the model to represent pedestrians’ lane formation and a more realistic pedestrian’s decision process;
- The inclusion of imprudent pedestrians in the simulation;
- The inclusion of cars and traffic lights in the simulation;
- The conceptual model proposed in this study to help the understanding of pedestrians’ crossings could be used to quantify how the provision of adequate systems for pedestrians can improve the use of these structures and, consequently, the safety of all users.

REFERENCES


Ronald, N. and Sterling, L. 2005. “Modeling pedestrian behavior using the BDI architecture”. International Conference on Intelligent Agent Technology (IAT’05), IEEE.


ACKNOWLEDGMENTS

The authors want to thank the Brazilian National Counsel of Technological and Scientific Development (CNPQ) for the financial support for the development of this research.

AUTHORS BIOGRAPHIES

CARLOS O. PRETTO was born in Santa Maria, Brazil and went to the Pontifical Catholic University, where he studied Automation and Control Engineering and obtained his degree in 2001. He received his Master’s degree in Electrical Engineering from the same university in 2005. Currently, he is a doctoring candidate at Laboratory for Transport System at the Federal University of Rio Grande do Sul. He is now working with agent-based simulation and pedestrian simulation.

HELENA B.B CYBIS was born in Porto alegre, Brazil and went to the Federal University of Rio Grande do Sul, where she studied Civil Engineering. She obtained her degree in 1980. She has a MSc (1989) and PhD (1993) from the Institute for Transport Studies University of Leeds,UK. Currently, she is a professor at the Laboratory for Transport System at the Federal University of Rio Grande do Sul.

PAULA ARIOTTI was born in Bento Gonçalves, Brazil and went to the Federal University of Rio Grande do Sul, where she studied Civil Engineering and obtained his degree in 2003. She received her Master degree in Production Engineering from the same university in 2006. Currently, she is a doctoring candidate at the Laboratory for Transport System at Federal University of Rio Grande do Sul. She is now working with pedestrians’ safety and vehicular emissions.
An agent-based simulation framework for modelling travel behaviour in Flanders

Katrien Ramaekers, Tom Bellemans, Davy Janssens and Geert Wets
Transportation Research Institute
Hasselt University
Wetenschapspark 5 bus 6, 3590 Diepenbeek, Belgium
e-mail: {katrien.ramaekers;tom.bellemans;davy.janssens;geert.wets}@uhasselt.be

Keywords: agent-based simulation, activity-based modelling, travel behaviour, Feathers

ABSTRACT

Modelling traffic patterns has always been a major area of concern in transportation research. A custom agent-based simulation framework is developed to model travel behaviour in the region of Flanders (Belgium). The framework uses an activity-based approach to model traffic demand. The major idea behind activity-based models is that travel demand is derived from the activities that individuals and households need or wish to perform. Activity-based models lead to more realistic traffic forecasts and as a result of this, the impact of transport on environment, traffic safety and health can be predicted more accurately. Activity data for the framework is provided by a large scale survey, conducted on 2500 households in the study area. The agent-based simulation model consists of over six million agents, one for each inhabitant, to represent the Flemish population. A modular design is used in order to manage complexity easily and to update and/or replace functionalities easily.

INTRODUCTION

Modelling traffic patterns has always been a major area of concern in transportation research. Since 1950, due to the rapid increase in car ownership and car use in the US and in Western Europe; several models of transport mode, route choice and destination have been used by transportation planners. These models were necessary to predict travel demand on the long run and to support investment decisions in new road infrastructure that originated from this increased level of car use.

The trip-based approach: the four-step model

The four-step model (Ruiter and Ben-Akiva 1978) has been the primary tool for forecasting future demand of regional transportation services. The four steps that can be distinguished are: trip generation, trip distribution, trip mode choice and trip assignment. The four-step model was introduced piece-wise in the late 1950s, and was significantly modified since its first implementations. Despite these modifications, the model still hangs on to the original standard framework.

The major advantage is the simplification that is incorporated into these models, which made urban passenger travel demand forecasting relatively easy. The simplifying assumptions facilitated the quantitative analysis of travel demand, while this is in fact a result of complex travel behaviour.

However, many of these aggregate four-step models show serious drawbacks (Jovicic 2001, McNally 2000). The major drawback clearly is the focus on individual trips, where the spatial and temporal interrelationships between all trips and the characteristics of trips are ignored. Furthermore, the overall behaviour is represented as a range of constraints that define transport choice, while in fact it is an outcome of both real human decision making and a complex choice process. The last drawback clearly is the complete negation of travel as a demand derived from activity participation decisions.

The tour-based approach

The original four-step models were replaced by theories about utility-maximising behaviour and individual choice behaviour. Multinomial logit models and more sophisticated techniques such as the nested logit and probit models formed the core of transportation modelling practice from the mid seventies onwards. Most of these techniques were implemented in so-called tour-based systems (Daly et al. 1983). In the tour-based model, trips are explicitly connected in chains that start and end at the same home or work base. This is done by introducing spatial constraints and by directions of movement. By means of this property, the lack of the spatial interrelationship, which was so apparent in the four-step trip based models, is dealt with. It is undisputable that much progress has been made in this research area compared to the aggregated four-step models. Nevertheless, these models did not escape criticism either. Especially in the eighties and early nineties, it was claimed by several researchers that very limited insight was offered into the relationship between travel and non-travel aspects. Indeed, travel has an isolated existence in these models and the question why people
undertake trips is completely neglected. This is where activity-based travel demand comes into play.

**The activity-based approach**

The major idea behind activity-based models is that travel demand is derived from the activities that individuals and households need or wish to perform. Travel is merely seen as just one of the attributes. Moreover, decisions with respect to travel are driven by a collection of activities that form an agenda for participation. Travel should therefore be modelled within the context of the entire agenda, or in other words, as a component of an activity scheduling decision. The concept of activity scheduling is an important one.

In short, travel patterns are the manifestation of the implementation of activity programs over time and space. In turn, activity patterns emerge as the interplay between the institutional context, the urban/physical environment, the transportation system and individuals’ and households’ needs to realize particular goals in life and to pursue activities (Ben-Akiva and Bowman 1998).

At least some of this complexity of travel decisions should be captured to make transportation models more reliable.

Activity-based approaches to transportation forecasting therefore aim at predicting which activities are conducted where, when, for how long, with whom, the transport mode involved and ideally also the implied route decisions.

Activity-based models lead to more realistic traffic forecasts and as a result of this, the impact of transport on environment, traffic safety and health can be predicted more accurately. However, despite these advantages, activity-based transportation models were not adopted at the same pace by practitioners due to their complexity and data requirements.

Over the last decade, several micro-simulation models of activity-travel demand (e.g. Cemadep (Bhat et al. 2004), Famos (Pendyala et al. 2005) and Albatross (Ar- entze and Timmermans 2000; 2005)) have become operational. One of the main challenges faced by the travel demand forecasting industry is the ability to rapidly deploy several new theoretical advances in a time and cost efficient manner. Therefore, it is of utmost importance to rely upon a sound basic platform where several of these advancements can serve as add-ons. Taking this into account, the idea was conceived in Flanders to develop a modular activity-based model of transport demand.

In this paper, the agent-based framework is presented to model travel behaviour in Flanders, using an activity-based approach. The agent-based simulation framework that is developed has the acronym Feathers (Forecasting Evolutionary Activity-Travel of Households and their Environmental RepercussionS). In the following section, the Feathers framework is described in more detail.

**THE FEATHERS FRAMEWORK**

The Feathers framework is a custom agent-based travel demand simulation framework. The agent-based approach makes it possible to simulate the behaviour of individual agents and their interactions with other agents explicitly. The overall behaviour of the system is formed by the cumulative effects of these individual behaviours. The units of investigation in an activity-based model are the persons making scheduling decisions that result in activity-travel diaries. Hence, the agents in the agent-based activity-based model are the individual persons.

To have a real-life representation of Flanders, the agent-based simulation model consists of over six million agents, each agent representing one member of the Flemish population. For each agent, a schedule of activities and journeys is drawn up. The scheduling model that is implemented in the Feathers framework for the simulations in the scope of this paper is based on the scheduling model that is present in the Albatross model (Arentze and Timmermans 2000). The scheduling is static and is based on decision trees. The data that is used to train these decision trees, is described in the next section. A sequence of 27 decision trees is used in the scheduling process. Decisions are made based on a number of attributes of the individual (e.g., age, gender), of the household (e.g., number of cars) and of the geographical zone (e.g., population density, number of shops). For each agent with its specific attributes, it is for example decided if an activity is performed. Subsequently, amongst others, the location, transport mode and duration are determined, taking into account the attributes of the individual.

Based on the individual schedules of all the agents, Feathers computes an origin-destination matrix. This origin-destination matrix aggregates all journeys in traffic demands between geographic zones. The origin-destination matrix serves as the input for the traffic assignment algorithm. The activity-based travel demand model of Feathers includes several modes of transport. Origin-destination matrices are computed for each mode and for different times of day (on- and off-peak conditions). However, in the traffic assignment model, incorporated in the Feathers framework, the focus is on the car mode. The origin-destination matrix is assigned to the road network of Flanders. A wide variety of traffic assignment methods exist. A well-known assignment method is the All-or-Nothing assignment, which uses a shortest path method to assign traffic to the network. This method, however, ignores the fact that link travel times are flow dependent. More advanced assignment methods take into account this volume dependence of travel times. Despite the additional computational burden, these more advanced models lead to more accurate results and are therefore preferable. The Feathers framework uses one of the more advanced methods: the equilibrium method. This method uses a speed-intensity
curve to model the impact of the link intensity on the speed. The information on the speed-intensity curves to use for the links is extracted from the Flemish multimodal model. This multimodal model is maintained by the Flemish government and is calibrated using real-life traffic counts. It is clear that aggregating the traffic demands resulting for all individuals in OD matrices yields an enormous loss of information. Therefore, in future work, a full agent-based micro-simulation framework will be developed to assign each agent individually to the network.

Data and data collection

Traditionally, travel surveys have been collected by paper and pencil or over the phone. The coming of activity-based analysis, which prompted the need for considerably more detailed data on travel behavior, identified the advantages of collecting activity or time use diary data (e.g., Clarke et al. (1981), see Ettema et al. (1997) for an overview). At the same time, however, the use of diary data virtually precluded the use of telephone interviews and in addition substantially increased respondent burden and error proneness, e.g., Dowling and Colman (1995), Sun et al. (1995). To avoid such error or at least reduce it, computer assisted diary instruments were developed. An activity-travel diary survey tool, called PARROTS (PDA (Personal Digital Assistant) system for Activity Registration and Recording of Travel Scheduling) was developed (Kochan et al. 2006). PARROTS runs on a PDA and uses the Global Positioning System (GPS) to automatically record location data. The PDA was programmed such that besides automatically registering its location, respondents can provide information about their activity-travel behavior as well. Whenever an activity or trip is registered in PARROTS, a number of attributes for this activity or trip are collected using a customized GUI. The most important activity and trip attributes PARROTS collects are: activity type, date, start and end time, location, mode of transportation, travel time and travel party. In order to collect the required data for building an activity-based model for Flanders, a large scale survey is being conducted on 2500 households. A paper-and-pencil survey and PARROTS are both being used on half of the surveyed households. The survey is conducted on households since the household context in which individuals operate has a very strong influence on individuals’ decisions, particularly when household resources are shared, there are shared household responsibilities and there are decisions that are made jointly by multiple household members. Besides the survey data, the Feathers framework also contains demographic and socio-economic geographical information about Flanders, such as age, gender, number of inhabitants, employment level, and the number of shops in geographic zones as well as data on the transportation networks. This data is available in the framework and constitutes the context in which the agent-based simulation is run.

Modular design

Feathers uses a modular design. Two important advantages of this modular design are the easiness to manage complexity and the easiness to update and/or replace functionalities as the state-of-the-art in the activity-based research field progresses rapidly. In order to be able to exploit Feathers’ modular structure to the maximum extent, a flexible configuration functionality is required. Every module that is active in FEATHERS communicates with the configuration module in order to obtain its specific required settings. One of the core modules in the system is the data module. The data module provides access to the data that needs to be accessible throughout all other modules. Two major types of data are provided by the data module: supply and demand data. The supply data consists of the data describing the ‘context’ in which the agents live and schedule their activity and travel episodes. This data not only includes the transportation network but also includes information on geographical zones in the study area such as attractiveness of a zone for conducting certain activities. Also information on the availability and performance of the transportation system between the zones in the study area (e.g. travel times, travel costs, bus fares) is included in the geographic supply data. The demand data consists of the activity-travel diaries or schedules that describe the demand for the execution of activities at certain locations as well as the resulting demand for transportation. The collected diaries are typically accompanied by person and household data for the persons executing the diaries. The data model for the demand data is aware of the following entities: persons, households, cars (optional), activities, journeys and lags. In order to perform a simulation of activity and travel behaviour of individuals in a population, a synthetic population consisting of persons and households (and optionally cars belonging to the household) needs to be built. The population module is responsible for the management of the different agents (persons) that are used in the synthetic population. The synthetic population therefore consists of a collection of agents where each agent is characterised by a number of attributes. The data required are available at population level in Flanders by means of the socio-economic survey. The schedule module is a generic module in which different scheduling algorithms can be implemented. At the moment, a decision tree-based scheduling algorithm was implemented in the schedule module. This implementation currently consists of a sequence of 27 decision trees, where each decision tree is used to model decisions.
on specific activity-travel schedule properties. Besides the decision trees, the scheduling mechanism contains an algorithm to make the schedules consistent. In order to be consistent, a schedule needs to comply with a number of constraints: situational constraints, institutional constraints, household constraints, spatial constraints, time constraints and spatial-temporal constraints. The output of the scheduler in the scheduling module is the collection of activity-travel diaries for all the agents in the population module.

The statistics module provides reports regarding the (synthetic) population and the activity-travel schedules to the Feathers user. As the activity-travel diaries contain detailed travel information, the statistical module also provides the functionality to compile an OD matrix. These OD matrices can then be assigned to the transportation network using the traffic assignment methods mentioned earlier.

CONCLUSIONS AND FURTHER RESEARCH

In this paper, an agent-based simulation framework is presented to model travel behaviour in Flanders. The main contribution of this framework is the application of an activity-based model on a large scale. A large scale data collection is being conducted using the Parrots tool. A modular design is used in order to allow for easy updating and replacement of functionalities.

In future work, the focus is on a tighter integration between the activity-based traffic demand model and the traffic assignment. Since aggregating traffic demands resulting for all individuals in OD matrices yields an enormous loss of information, the goal is to develop a full agent-based micro-simulation framework to assign each agent individually to the network instead of using OD matrices.

Based on the detailed routing information from Parrots, it is possible to look for a relationship between the context of a trip and the chosen route. It is assumed that the context of a trip plays a role in how an individual chooses a certain route. Using the information from Parrots, the individual assignment of each agent to the network can take into account the context of the trip.

Another goal in future work is the development of a module for freight transport in the Feathers framework. Since freight transport is an important part of total traffic on the Flemish network, it is of great importance that it is added to the model. Due to the modular design of the Feathers framework, an extra module can be added to the framework without too much extra effort.

REFERENCES


A General Framework for Routing Problems in Real Traffic Conditions

Michelle Chabrol, Michel Gourgand and Patrice LeClaire
LIMOS, UMR 6158 CNRS, Complexe Scientifique des Cézeaux 63173 AUBIERE, France
email: {chabrol,gourgand,leclaire}@isima.fr

KEYWORDS
Routing Problems, Urban Traffic System, Double complexity, Module coupling, Optimization, Simulation

ABSTRACT

We propose in this paper to study routing problems in real traffic conditions. These problems have a double complexity arising from the algorithmic complexity of the problems and the systemic complexity of the underlying system. The evaluation of some of their criteria cannot be performed analytically but via simulation. To handle with the double complexity of these problems, we use the modelling methodology ASDI in order to produce a generic knowledge model of these problems and of the system on which they take place. The proposed analysis scheme provides all the elements useful to process a module coupling between an optimization method and a simulation model. This methodology is applied to the Routing Problems in Real Traffic Conditions (RP-RTC).

INTRODUCTION

The routing problems are most of the times studied on roads on which the vehicles go without considering interactions with the other vehicles. There is no source of stochasticity in this kind of problems. The study of these routing problems in real traffic conditions is a new research domain which emerges with the recent development of the new technologies of communication such as GPS and WIFI protocol, bringing real-time information on traffic conditions more available. The complexity of this kind of problems is the result of an algorithmic complexity (of the routing problems) and a systemic complexity (from the underlying system on which the problems take place). The classic optimization methods do not manage these problems, because the evaluation of some criteria cannot be performed analytically, but via simulation. We propose in this paper a modelling methodology which includes the double complexity of such a class of problems and the possibility to easily perform module coupling. We first present the methodology for any kind of problem class having double complexity, then apply it to the class of vehicle routing problems in real traffic conditions.

This paper is organized as follows. In a first section, we present the notion of double complexity. The second section is devoted to the coupling optimization-simulation and focuses on the case in which the optimization method controls the simulation model. Then, in a third section, we present the used modelling methodology, ASDI (for Analysis, Specification, Design and Implementation) and the improvements proposed to deal with the double complexity. The fourth section is on the application of this methodology to the class of Routing Problems in Real Traffic Conditions (RP-RTC), providing the corresponding generic knowledge model and an example of possible module coupling between meta-heuristic and simulation for a problem of this class.

DOUBLE COMPLEXITY

In literature, the studied problems can be divided into two types according to the concerned system: to-be-conceived systems or to-be-optimized existing systems. The managers in charge of proposing solutions for these problems can use tools and methods for the evaluation of the performances of their existing system, solve dimensioning problems for a new system, or even obtain a robust functioning of their system under stochastic events. The problems are optimization problems and/or evaluation problems.

Concept of double complexity

For some classes of problems, Norre (2005) defines the double complexity as follows:

- A systemic complexity, resulting from the difficulty in simply evaluating one or several performance criteria of the system.

- An algorithmic complexity, due to the complexity of the combinatorial optimization problems, in which one seeks a solution that optimizes one or several performance criteria. The combinatorial optimization problems are in general NP-complete problems.

Figure 1 represents the double complexity. At the intersection of the two ellipses are the optimization problems for which performance criteria cannot be simply evaluated.
Solving methods

In order to deal with the algorithmic complexity, one can use a large number of operational research methods divided into two categories: exact methods and heuristic methods. For large instance problems, exact methods cannot often be applied, whereas heuristic and meta-heuristic methods are useable.

In order to evaluate the performance of a system, simulation or Markovian models can be used.

The coupling of optimization methods and performance evaluation models allows to solve problems with double complexity.

COUPLING OPTIMIZATION-SIMULATION

In the literature

The coupling of methods is a simple concept taking advantages of two or more methods or performance evaluation models. The following definition was taken from Norre (2005).

The coupling of two modules $M_1$ and $M_2$ (figure 2) is defined as the running of module $M_2$ inside module $M_1$.

The coupling of both modules can consist in running $M_2$ at every iteration of $M_1$ in order to improve the solution proposed by $M_1$ or to evaluate some criteria by $M_2$. The coupling of modules $M_1$ and $M_2$ ($M_2$ run inside $M_1$) is noted $M_1 \hookrightarrow M_2$.

<table>
<thead>
<tr>
<th>Module $M_1$</th>
<th>demand of $M_1$</th>
<th>Module $M_2$</th>
<th>response of $M_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>begin of $M_1 \hookrightarrow M_2$</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>end of $M_1 \hookrightarrow M_2$</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

Figure 2: Coupling of $M_1$ and $M_2$

$M_1$ and $M_2$ can be optimization methods and/or evaluation models (simulation, Markovian models).

Simulation optimization

This section focuses on a specific case of module coupling in which $M_2$ is a simulation model. The term 'simulation optimization' is used in the literature, for instance by Carson and Maria (1997) and Fu et al. (2005) for this case.

The simulation is used to evaluate the performance of a system under a given number of structural, organizational and/or operational assumptions, in order to answer to 'what if' questions, for certain value of the parameters of the studied system.

The simulation optimization methods provide a framework that allows to answer to 'how to' questions by finding out the optimal parameters of a given simulation model.


The aim of a simulation optimization is to minimize the objective function:

$$\min_{\theta \in \Theta} J(\theta) = E[L(\theta, \omega)]$$

where $\theta \in \Theta$ represents the input variables, $J(\theta)$ is the objective function, $\omega$ represents a simulation replication, and $L$ is the replication performance measure.

The complexity of the simulation optimization lies in the estimated evaluation of the performance of a system, which does not guarantee the comparison of two configurations. Thus, it is necessary to realize several replications. The cost of the optimization depends on the number of replications used to evaluate the objective function. This number of replications can vary according to the value of $\theta$, i.e. the parameter set, and for every iteration of the search process.

Integration in a software

Most of the cited above surveys on simulation optimization integrates a practical part describing the integration of such a method in the commercial simulation software.

Fu (2002a) noticed that simulation software integrate some optimization routines to optimize the user-selected parameters according to chosen performance measures. But the user cannot know when the optimum is reached. Moreover, the coupling is not so easy to realize.

Ölaufsson and Kim (2002) remarked that most of the times the optimization method used is a metaheuristic, even if such a method does not guarantee strict convergence. However, these methods are still enough fast and robust for a practical use.
Fu (2002a) gave the characteristics of a good implementation of a simulation optimization: generality (for a functioning with a large number of problems), transparency to the user, high dimensionality (possibility to deal with large instances), efficiency in terms of computational time and busy percentage of the computational resources, possibility to integrate a set of standard problems.

To deal with the double complexity, we need a modelling environment allowing to model a problem with double complexity and develop module couplings using a simulation model.

**MODELLING METHODOLOGY FOR COUPLING OPTIMIZATION AND SIMULATION**

A modelling environment is constituted by a package of methods and tools including a performance evaluation software based on simulation, a graphical layer, a result exploitation layer, a decision-making layer providing dashboards for data analysis, a data layer, a statistic and optimization layer, an analysis and specification layer, a domain modelling methodology layer for the elaboration of simulation models.

The ASDI methodology (Analysis - Specification - Design - Implementation), developed by Gourgand and Kellert (1992), describes the framework of a software environment for modelling and simulation. This methodology gives some elements for the design and/or management of complex systems and recommends the re-use of the knowledge of a domain for the study of the systems which compose it; a domain can be defined as a class of systems with common technical and functional characteristics. The main contribution of this methodology consists in separating the knowledge gathering from the construction of the computer model, so as not to depend on a particular programming language and to promote the reusability of the components as much as possible.

The knowledge model describes the structure and the functioning of the system in a natural or graphic language (UML, Aris for instance), thanks to a decomposition in three subsystems:

- The logical subsystem (LSS): elementary services that the system have to treat,
- The physical subsystem (PSS): physical entities, their geographical repartition and their interconnections,
- The decision subsystem (DSS): management rules of the interactions between LSS and PSS.

This model is then translated in the action model thanks to a mathematical and/or programming language. Several action models can be obtained from the same knowledge model. Chabrol et al. (2006) have proposed an extension of this methodology, called ASDI-mi, so as to integrate the multiple and incremental aspects in modelling. ASDI-mi allows to deal with the structural and functional complexity of a system step by step, to study it according to various horizon lengths, to combine various action models, and even various models of different levels of abstraction.

**Double complexity modelling**

We propose an improvement of the ASDI-mi methodology to deal with the double complexity of some problems (figure 3).

![Figure 3: Overview of the proposed methodology for the double complexity problem class](image)

As mentioned before, the first step of the ASDI methodology is to produce a generic knowledge model of the studied domain. In the case of double complexity, this analysis of the domain is more complex and requires a double competence from the designers: they need to be expert in simulation, *i.e.* in the modelling of the studied system and in operational research in order to have a certain knowledge level on the optimization problems that can take place in the system.

The elaboration of the generic knowledge model can be done, for instance in UML, by describing all the necessary entities for:

- the simulation of the system: decomposition in three subsystems, giving the physical, logical and decisional views of the system,
- the optimization process: integration of the additional attributes or classes used to allow a future complete description of the possible problems in terms of objective functions, decision variables and constraints.
Once the generic knowledge model is obtained for the domain, we can specify the studied problem and the underlying system. This instanciation to a specific system allows to obtain a sharper description of the problem and the system, thus reducing the useful part of the generic knowledge model to the strict requirements of both optimization and simulation.

This knowledge model of the studied system can then be derivated into an action model according to the to-be-solved problem, the to-be-applied method and the chosen programming language. With the complete description of the system with UML (or any other modelling language) diagrams understandable by any modelling expert, the derivation into a given action model does not bring any other source of difficulty. The method realizing the coupling of an optimization method and the simulation model will use a double derivation of the classes of the knowledge model: first the classes useful for the simulation (with the creation of some new attributes dedicated to the simulation process), then the attributes (and, if necessary, the classes) useful for the optimization method.

To bring more reusability to the overall process, one can think to develop a component library providing an interface for the use and configuration of the standard metaheuristics with the most common neighbourhood operators. Their utilization need adaptations for a given system.

APPLICATION TO THE ROUTING PROBLEMS IN REAL TRAFFIC CONDITIONS

In this section, we focus on the routing problems in real traffic conditions. First, we describe the source of the double complexity of such a problem, and then use the methodology to produce the generic knowledge model for the RP-RTC domain, the knowledge model of a specific problem of this domain, and finally an action model using the coupling between an optimization method and a simulation model. We used the RP notation, in order to not mistake it with the classic VRP, which is a specific problem of the RP domain.

Double complexity for the routing problem class

The routing problems have been studied extensively throughout the past decades. The first problem is the Travelling Salesman Problem (TSP) which aims at finding an Hamiltonian path on a non-oriented graph in which the nodes are the cities to be visited and the edges are the links between cities. The Vehicle Routing Problem (VRP) is a multi-vehicle extension of the TSP with vehicle capacity constraints. For both problems, the aim is to deliver or collect some goods to or from the visited locations. The Pickup-and-Delivery Problem (PDP) models both actions, adding the pairing and precedence constraints. The goods are transported from a pickup (collect) location to a delivery location by a single vehicle. Some time windows constraints can be added to model the time interval in which the service (pickup and/or delivery) has to be done. Finally, with the Dial-A-Ride Problem (DARP), the carried entities are no more goods but people specifying time windows constraints. A survey of this class of routing problems can be found in Savelsbergh and Sol (1995). The authors proposed a general model applicable to all the problems of this class they called the General Pick-up-and-Delivery Problem (GPDP). The TSP, VRP, PDP and DARP are considered as particular cases of the GPDP as a combination of a constraint set and a function objective. All these problems are known to be NP-complete as they are all an extension of the TSP which is itself NP-complete. This is the algorithmic complexity.

Another important research community is interested in the class of the Urban Traffic Systems (UTS). For these systems, several models can be found according to the decision level (strategic, tactic or operational) and to the considered abstraction level (macroscopic, mesoscopic or microscopic). Figure 4 illustrates the different kinds of problems one can deal with, for given decision and abstraction levels. The mesoscopic abstraction level is an intermediate state between the macroscopic and microscopic ones, allowing for example to obtain results more rapidly with a lower level of details compared to the microscopic level. The more detailed the abstraction level is, the more time consuming the computation is. The complexity of such a system arises from the network and all its functioning rules: driver behaviour, lane changing, intersection, priority rules, traffic light functioning, speed limitation, speed management (acceleration, braking). This is the systemic complexity.

![Figure 4: Examples of studies for the UTS class according to given decision and abstraction levels Sarramia (2002)](image)

An extensive literature deals with the vehicle routing problems under the simplifying assumption that the vehicles move on dedicated roads with fixed travel times.
Fu (2002b) integrated stochasticity in the travel times according to historical data. Fu (2002c) proposed a simulation model which integrates the traffic conditions in the travel times without explicitly considering traffic flow in detail.

We focus in this paper on the routing problems in real traffic conditions. The double complexity (figure 5) comes from the algorithmic complexity of the vehicle routing problems and the systemic complexity of the network on which they take place. For this kind of problems, we use the operational decision level.

Figure 5: Double complexity for the class of routing problems on real traffic conditions

In the next part, we apply the methodology above described on the domain of the Routing Problems in Real Traffic Conditions (RP-RTC) we identified like having a double complexity.

**Generic knowledge model for the RP-RTC domain**

For the modelling of the underlying system, we used the work of Sarramia (2002) who described the generic knowledge model for the domain of urban traffic systems.

Our contribution consists in integrating the modelling of the particular requirements of the class of routing problems. To model a general routing problem, we have to gather informations on the most common characteristics, constraints and function objectives of these problems.

To give a description of the functioning of such a problem, we can quote the example of the General Pickup-and-Delivery Problem (Savelsbergh and Sol (1995)): from a fleet of vehicles of given capacities, the GPDP consists in satisfying as many transportation requests as possible. A request is emitted by a customer who specifies the origin and/or destination location(s) and the corresponding desired service times (thus providing time windows in which the service has to be performed). The classical constraints of the GPDP can be the capacity, pairing, precedence, maximum travel time, maximum ride time and time window constraints. The most frequently objective functions are the minimization of the operational costs or the maximization of the quality of the service.

Desrochers et al. (1990), and more recently Berbeglia et al. (2007) and Haj Rachid et al. (2008) (from the VeRo & ClEo – Vehicle Routing: Classification and Evolutionary Operators – project) proposed a classification of the routing problems. The classification of Haj Rachid et al. (2008) uses four fields $\pi/\alpha/\beta/\gamma$:

- $\pi$ – Scheduling horizon.
- $\alpha$ – Resources:
  - Fixed resources: depot type (backhaul, linehaul), number of depots, graph type (directed or not).
  - Movable resources: number of vehicles, fleet type (homogeneous or not), number of workers (and their abilities).
  - Requests: location (nodes and/or edges), customer type (pickup and/or delivery), number of products or customer types, static or dynamic nature of the demand, static or dynamic costs of service and transport.
- $\beta$ – Constraints: capacity, autonomy, time windows, number of serviced clients, traffic conditions, split delivery, multi-trips, come back to the initial depot, departure depot, service frequency, minimum or maximum interval between two services, departure depot in the multiperiod case, pairing, precedence, compatibilities.
- $\gamma$ – Objective functions: minimization of the total distance, of the number of vehicles used, of the total waiting time, of the unserviced customers, of the service time, of the total time, of the dissatisfaction of the customer.

De Paepe et al. (2004) developed a software for the automatic classification of different types of DARPs according to the types of vehicles, travels, metric spaces and objective functions.

Such classifications, generic models like the GPDP, and a study of some specific problems provides us with sufficient knowledge to model a generic vehicle routing problem. Figures 6, 7 and 8 are the class diagrams respectively associated to the physical, logical and decisional subsystems of the generic knowledge model for the RP-RTC domain. We complete the generic knowledge model of the urban traffic system of Sarramia (2002) with the shaded classes, which are devoted to the routing problems.

The Vehicle class in the PSS represents the vehicles of the routing problems whereas the MeansOfTransport class in the LSS is devoted to the moving entities of the urban traffic system simulation model. The Vehicle class has some attributes such as capacity and autonomy. The Client class corresponds to the requests emitted by a customer, so that it contains attributes like the...
Figure 6: Class diagram of the PSS

Figure 7: Class diagram of the LSS

Figure 8: Class diagram of the DSS
time windows, the to-be-transported people. The times of service of a given solution can be saved in the Pick-upOrDeliveryStop class which is associated to an Origin or a Destination of the Route assigned to the client Request(Flow). The CompleteDepot class represents a possible multiple heritage to model a facility combining the functioning of a Depot, a Petrol service station and a product Warehouse. DC and SC are abbreviation respectively for DecisionCenter and SchedulingCenter. Figure 9 is a class diagram representing some links between the three subsystems. We just focus on the new part concerning the routing problems.

Figure 9: Links between the three subsystems

Knowledge model for a specific domain

The passage from the generic knowledge model of the RP-RTC domain to the knowledge model of a specific domain such as the TSP-RTC, VRP-RTC, PDP-RTC or DARP-RTC uses the concept of filtering. Let us consider the DARP in Real Traffic Conditions (DARP-RTC). The knowledge model of the DARP-RTC contains the strictly necessary classes for the description of the specific domain. A specific system will be an instance relatively to one of these specific domains. The classic DARP can be expressed as follows: from a homogeneous fleet of vehicles, the DARP consists in satisfying as many transportation requests as possible. A request is emitted by a customer who specifies the origin and destination locations and the time windows for the corresponding desired service times. The classical constraints of the DARP are the capacity, pairing, precedence, maximum travel time, maximum ride time and time window constraints. One can take the minimization of the dissatisfaction of the clients as a objective function. The different problems found in the literature vary according to the objective function, some constraints like the definition of only one time window for instance or the adding of some extension, which are already modelled in the generic knowledge model of the RP-RTC domain. Some surveys on the DARP can be found in Cordeau et al. (2006), Cordeau et al. (2004) and Parragh et al. (2008).

The PSS, LSS and DSS class diagrams of the DARP-RTC domain contains only the classes and links necessary to the above description of the problem. This specific knowledge model is obtained by filtering the generic one, i.e. choosing the desired classes, levels in the classes hierarchies and corresponding links.

Action model

Once the knowledge model of a specific domain is obtained, the next step is the derivation into an action model according to some choices concerning the methods to be used. In the case of coupling of two methods, a two-phase derivation is necessary.

In the case of the DARP-RTC, the evaluation of the travel times must be realized through simulation. A solving method can be a coupling between a metaheuristic and a simulation model. At each iteration of the metaheuristic, a new solution is obtained thanks to an operator providing a neighbour of the current solution, and then sent to the simulation model which returns back the evaluation of the criterion (for instance, the evaluation of the travel times). The process will use several replications at each iterations.

Figure 10: Coupling of metaheuristic and simulation for the DARP-RTC

For the derivation of the simulation part of the coupling, we use the work of Sarramia (2002) who used a classic decomposition in three models: a representation model, a flow model and an assignment model. These models allows to obtain a realistic behaviour of the moves of the vehicles in the detailed network.

To this implementation, some data for each vehicle of the fleet of the DARP has to be added in order to compute the statistic results. The evaluation of the objective function of the metaheuristic can be delegated to the simulation module.

For the derivation of the metaheuristic part of the coupling, we need to define the parameters of the metaheuristic (the number of parameters depends on the type of the metaheuristic), the neighbourhood operator and the underlying data structure. An insertion operator can easily be implemented by modifying the connection between a request and a vehicle to which the request is assigned. No additional structure is needed with this operator.
CONCLUSION

We present in this paper a modelling methodology for the problem class with double complexity. This double complexity arises from the algorithmic complexity of a problem and the systemic complexity of the underlying system. In this problem class, the evaluation of some criteria can only be performed via a simulation model. The proposed methodology allows to obtain a generic knowledge model of a problem class in its underlying system and one or several action models using the module coupling. The solving methods combine optimization methods and simulation models.

We applied this methodology to the Routing Problems in Real Traffic Conditions (RP-RTC) and proposed a generic knowledge model allowing to develop an action model using module coupling for any routing problem. In prospect, one can think to apply the methodology to a real instance and study deeper the module coupling proposed and how to accelerate the evaluation of the replications.

REFERENCES


Norre S., 2005. Heuristiques et Méthodologies pour la Résolution de Problèmes d’Optimisation Combinatoire dans les Systèmes de Production.


A TOOL FOR CAR TRAFFIC MANAGEMENT

Alfredo Cutolo
Raffaele Frattarulo
Luigi Rarità
Department of Information Engineering and Applied Mathematics
University of Salerno
84084, Fisciano (Salerno), Italy
E-mail: {cutolo, frattarulo, rarita}@diima.unisa.it

KEYWORDS
Simulation tools, traffic flows, optimization.

ABSTRACT
A numerical tool for the macroscopic simulation and optimization of car traffic is presented. A user-friendly interface allows the construction of the topology of road networks. The tool can be used to study the traffic evolution on road networks, and to suggest some interventions to avoid congestion phenomena.

INTRODUCTION
Nowadays, the exponentially increasing number of circulating cars along urban networks makes the problem of traffic control very difficult. The growing interest of scientific and industrial community toward modelling of car traffic on road networks in critical situations produced various simulators, whose aim is an accurate description of the evolution of traffic flows, and the possibility to test the results of some changes before deploying it in a real system. Traffic simulators can be classified as microscopic and macroscopic. The first ones consider individual vehicles, while the second ones simulate the overall flow of traffic, resulting from the interaction of a lot of cars, and then showing the traffic network from a bird’s eye.

Among simulators based on microscopic variables we cite CORSIM/TSIS (see Corsim and Owen et alii 2000), MITSIM (Yang et alii 2006), PARAMICS (Cameron et alii 1994), VATSIM (Redmill et alii 2001) RENAISSANCE (Messmer et alii 2006), which uses discrete data, is the only application that models the traffic macroscopically. FreeSim (Miller et alii 2007) and VISSIM (Fellendorf 1994) both allow for macroscopic and microscopic vehicles evolution.

The aim of this paper is to present a tool for the management of car traffic flows on networks, based on a fluid-dynamic macroscopic model for car networks suggested by Cocilite et alii 2005 and Garavello et alii 2006. Such tool, realized by a core in C++ and a graphical interface in Java, is able to simulate in a dynamic way the considered network and, in particular, situations of congestions derived by car accidents, giving as input the starting scenario and initial/boundary conditions on roads of the network.

Few data and parameters are needed to realize time-continuous simulation of big car traffic networks, consisting of thousands and thousands of nodes and arcs, with not expensive computational times.

An important feature of the simulative tool for car traffic is the integration of routines of traffic densities with optimization procedures (Cascone et alii 2007, Cascone et alii 2008), that can indicate how car flows can be improved. In fact, especially on urban networks, characterized by short roads and many junctions, congestion phenomena are very frequent. Once one has individuated the more critical point on a road network, the optimization routines are able to suggest the best intervention to alleviate congestions, in terms of traffic signals and temporization of traffic lights.

The paper is organized as follows: first, we present the simulative tool; then, we consider the analysis of a real network, which is in Pompei, Italy, in order to give some strategies for the improvement of car traffic.

SIMULATIVE TOOL

Simulative algorithm
The simulative algorithm for the car traffic tool is based on the idea to look at large scales so as to consider cars as small particles and their density as the main quantity to be considered. In this case, it is reasonable to assume the conservation of the number of cars, thus leading to a conservation law (for exhaustive explanations, see Cocilite et alii 2005 and Garavello et alii 2006). Thus, on each single road, the evolution model is governed by the scalar hyperbolic conservation law:

\[ \partial_t \rho + \partial_x f(\rho) = 0, \]  

where \( \rho = \rho(t,x) \in \mathbb{R} \), \( t \in \mathbb{R} \), is the density of cars, \( \rho_{\text{max}} \) is the maximal density of cars, \( f(\rho) = \rho v \) is the flux and \( v \) the average velocity. We further assume that \( v \) is a smooth decreasing function of the density \( \rho \) and \( f \) is concave. The behaviour of the flux is usually estimated by measures on the roads of a given network. At road junctions, the only conservation of cars (all the incoming flux is equal to the outgoing flux) is not sufficient to determine a unique solution for traffic. Hence, it is necessary to state some rules:

(A) there exists a traffic distribution matrix, whose coefficients \( \alpha_{ij} \), represent the probability that cars, from the incoming road \( i \), take the outgoing road \( j \);

(B) respecting (A), drivers behave so as to maximize the flux through the junction.
Moreover, if the number of incoming roads is greater than the number of outgoing roads, a yielding rule is needed. This corresponds to fix right of way parameters, which permit to find a unique solution. More precisely, the $i$ – th parameter indicates the percentage, among cars passing through the junction, coming from the $i$ – th coming road.

The described mathematical model must be treated numerically, using some numerical methods for partial differential equations (see Godunov 1959). The algorithm allows to simulate the evolution of the density on each arc of the network, for each segment of the arc and at each time, also considering the phenomena of traffic at junctions.

**Optimization routines**

The tool for car traffic is also characterized by some optimization routines, which establish what are the best interventions to make traffic decongested. In particular, according to the approach suggested in Cascone et alii 2007 and Cascone et alii 2008, the optimization considers right of way parameters. A suitable choice of such parameters corresponds to the adoption of traffic signals or an adequate temporization of traffic lights at road junctions. The correct choice, as for traffic signals and traffic lights, is made through the analysis of two cost functionals, $J_1$ and $J_2$, that measure, respectively, the average velocity and the average travelling time of cars, which travel along the roads of the network. The aim is the maximization of $J_1$ and the minimization of $J_2$, thus leading to an optimization algorithm, that gives suitable right of way parameters.

**Graphic interface**

The first thing that a user can do is to draw the topology of the network to simulate. The user can create nodes and arcs (with related directions of traffic flows), through a simple interface consisting of buttons, as we can see in Figure 1.

![Figure 1: Nodes and arcs defined by the user.](image)

Once the graph has been defined, according to the model the simulation parameters for each arc have to be inserted. In particular, for every arc and node, a menu is associated, which can be visualized clicking twice on them by the right button of the mouse. The menu for nodes consists of only one feature: *remove*, which allows to remove the selected node. Instead, the menu for arcs is characterized by two features: *remove*, by which it is possible to clean the selected arc, and *properties*, by which it is possible to insert the following characteristics for each arc (Figure 2):

- **Name**: name of the road;
- **Start Value**: initial density on the road;
- **Start Density**: incoming flux on the road (necessary as boundary data, if the node is virtual);
- **Precedence**: the right of way parameter (necessary if the road belongs to a junction with a number of incoming roads greater than the number of outgoing roads);
- **Distribution**: the distribution parameter, necessary to define traffic distribution matrix for the junction;
- **Length**: length of the road (in meters).

Once the parameters are inserted, it is also possible to label every road with its own name (Figure 3).

![Figure 2: Menu for parameters.](image)

![Figure 3: Labeling of roads.](image)

Moreover, the traffic tool has been planned in such way that it is possible to load a map related to the real topology of the network, which has to be simulated (Figure 4). The user can draw on the map the network to be analyzed, following the real topology of the network; or he can experiment some changes in the traffic graph, like the modification of a road from a bidirectional flow into a unidirectional one, or the introduction of a traffic light, or the creation of a new road in order to evaluate the consequent traffic conditions. Maps of the interested road network are auspically obtained by Google Earth. An example, that concerns the harbour of Salerno, Italy, is in Figure 5.

![Figure 4: Map of the network.](image)
car traffic. Once a critical node is found, the simulative tool modifies the map inserting pictures of traffic signals or traffic lights with related green and red phases times.

APPLICATION

In this section, we analyze the car traffic that interests a part of the city of Pompei, Italy. The network concerns a set of roads that are near the santuario, the Pompei big church dedicated to Holy Mary. As Pompei is a city characterized by a great amount of tourists, zones around the santuario are heavily interested by strong car flows. In Figure 6, we show the map of the road network, and in Figure 7, the corresponding configuration loaded by the simulation tool.

![Figure 6: Topology of the Pompei network around the santuario.](image)

![Figure 7: Topology of the Pompei network with map loaded by the tool.](image)

Table 1: scale of red for different intervals of cars densities.

<table>
<thead>
<tr>
<th>Density interval</th>
<th>Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 \leq \text{density} &lt; 0.1$</td>
<td></td>
</tr>
<tr>
<td>$0.1 \leq \text{density} &lt; 0.2$</td>
<td></td>
</tr>
<tr>
<td>$0.2 \leq \text{density} &lt; 0.3$</td>
<td></td>
</tr>
<tr>
<td>$0.3 \leq \text{density} &lt; 0.4$</td>
<td></td>
</tr>
<tr>
<td>$0.5 \leq \text{density} &lt; 0.6$</td>
<td></td>
</tr>
<tr>
<td>$0.7 \leq \text{density} &lt; 0.8$</td>
<td></td>
</tr>
<tr>
<td>$0.8 \leq \text{density} &lt; 0.9$</td>
<td></td>
</tr>
<tr>
<td>$0.9 \leq \text{density} &lt; 1$</td>
<td></td>
</tr>
</tbody>
</table>

Optimization routines are able, analyzing the various densities on the arcs of the network, to focus on a critical road junction, that presents some congestion phenomena, and to suggest the best intervention for the improvement of

![Figure 8: Estimated flux function vs density.](image)
The following table shows the parameters by which the Pompei network has been simulated. Notice that the incoming flux, that corresponds to the boundary data, is needed for roads that inject traffic from the external or for roads that let traffic flows go out of the network, for example roads Piave and Sacra. For such roads, the incoming flux is 0.5, while initial conditions for all roads are assumed to be zero.

Table 2: parameters of simulation.

<table>
<thead>
<tr>
<th>Name</th>
<th>Right of way</th>
<th>Distribution</th>
<th>Length (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piave</td>
<td>/</td>
<td>/</td>
<td>120</td>
</tr>
<tr>
<td>Astorello</td>
<td>/</td>
<td>0.2</td>
<td>60</td>
</tr>
<tr>
<td>BLongo</td>
<td>0.7</td>
<td>0.8</td>
<td>350</td>
</tr>
<tr>
<td>Sacra</td>
<td>0.3</td>
<td>/</td>
<td>250</td>
</tr>
<tr>
<td>BLongo2</td>
<td>0.6</td>
<td>/</td>
<td>80</td>
</tr>
<tr>
<td>ColleSanB</td>
<td>0.4</td>
<td>0.3</td>
<td>60</td>
</tr>
<tr>
<td>BLongo3</td>
<td>/</td>
<td>/</td>
<td>60</td>
</tr>
<tr>
<td>BLongo4</td>
<td>/</td>
<td>0.4</td>
<td>60</td>
</tr>
<tr>
<td>BLongo5</td>
<td>/</td>
<td>0.6</td>
<td>60</td>
</tr>
<tr>
<td>SanMichele2</td>
<td>/</td>
<td>0.5</td>
<td>60</td>
</tr>
<tr>
<td>BLongo6</td>
<td>0.3</td>
<td>0.5</td>
<td>60</td>
</tr>
<tr>
<td>Roma</td>
<td>/</td>
<td>/</td>
<td>100</td>
</tr>
<tr>
<td>Roma2</td>
<td>/</td>
<td>0.6</td>
<td>200</td>
</tr>
<tr>
<td>Roma3</td>
<td>/</td>
<td>0.3</td>
<td>300</td>
</tr>
<tr>
<td>VE2</td>
<td>0.7</td>
<td>0.7</td>
<td>160</td>
</tr>
<tr>
<td>VE</td>
<td>0.3</td>
<td>/</td>
<td>60</td>
</tr>
<tr>
<td>ColleSanB3</td>
<td>/</td>
<td>/</td>
<td>200</td>
</tr>
<tr>
<td>SanMichele</td>
<td>/</td>
<td>0.4</td>
<td>60</td>
</tr>
<tr>
<td>CollesanB2</td>
<td>0.7</td>
<td>0.7</td>
<td>160</td>
</tr>
</tbody>
</table>

With a choice of the total time of simulation equal to 15 minutes, in Figure 9 and 10, we have, respectively, the situation at 7 minutes and 15 minutes.

In Figure 9, you can notice that the whole network is not full again, but road Sacra already presents some congestion phenomena with respect to the other roads. A typical reverse example is given by road San Michele, which is almost empty.

In Figure 10, the traffic, coming from roads Piave and Sacra, is injected to all roads of the network. In particular, problems of congestions are on roads Sacra and ColleSanB3.

Figure 9: Situation after 7 minutes of simulation.

Figure 10: Situation after 15 minutes of simulation, namely the end of the simulation process.

In Figures 11, 12, and 13, we present some images, elaborated by the tool, concerning the density of traffic on the road Sacra. Figure 11 shows the behaviour of the density on road Sacra after 6 minutes. Notice that the congestion is propagating along the road. The situation is more remarked in Figure 12, where the situation on road Sacra is depicted after 6 minutes and 30 seconds. Figure 13, instead, shows the variation of the density at the middle of road Sacra (125 meters) during all the simulation. Such figure shows how the density, starting from the value zero (initial condition for road Sacra equal to zero at the beginning of the simulation), assumes very high values when time increases.

Figure 11: Density on road Sacra after 6 minutes.

Figure 12: Density on road Sacra after 6 minutes and 30 seconds.
The optimization routine, which characterizes the tool, indicates that, on the basis of the simulated traffic flows, the more congested road junction has roads BLongo, BLongo2 and Sacra. For such junction, an optimization of right of way parameters is needed. In Figure 14, it is shown the solution that the tool elaborates: the adoption of a traffic light. Several other possible solutions could have been given; for example, according to the entity of congestion phenomena, the congested junction would have been optimized through a suitable traffic signal, namely a stop sign or a right of way signal.

Table 3 shows the optimal parameters for the studied road junction, in terms of green and red phases for cars coming from road Sacra, according to various choices (decided by the network designers) of the traffic light total cycle.

The cost functional $J_1$ in the optimal configuration (red line), elaborated by the tool and the simulated configuration (green line), corresponding to the parameters of Table 2 are in Figure 15. The blue line in such Figure represents the simulation of the road junction if a stop sign is adopted along road Sacra. As we can see, the optimal configuration exhibits a better performance, as the cost functional is higher.

CONCLUSIONS

In this paper, we have presented a simulative tool able to analyze the evolution of car traffic flows in a road network. The potentialities of the tool have been used to simulate a part of the real urban network of Pompei, Italy. The use of optimization routines allow to establish what are the correct interventions for the improvement of car traffic, with consequent reduction of congestion phenomena.

REFERENCES


BIOGRAPHY

**ALFREDO CUTOLE** was born in Salerno, Italy, in 1980. He graduated in Informatics in 2005, with a thesis on a Geographical Information System (GIS). He is actually a PhD student in Mathematics at the University of Salerno. His scientific interests are about dynamic models for traffic flows on road and telecommunication networks. His e-mail address is cutolo@diima.unisa.it.

**RAFFAELLA FRATTARUOLO** was born in Foggia, Italy, in 1980. She graduated cum laude in Mathematics with a thesis on the numerical treatment of integral equations. She is actually a PhD student in Mathematics at the University of Salerno. Her scientific interests are about dynamic models for traffic flows on road and telecommunication networks. Her e-mail address is frattaruolo@diima.unisa.it.

**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, in particular tandem queueing networks with negative customers and blocking. He obtained PhD in Information Engineering in 2008 at the University of Salerno. He is actually a research assistant at the University of Salerno. His scientific interests are about queueing theory, numerical schemes and optimization techniques for fluid – dynamic models. His e-mail address is rarita@diima.unisa.it.
COLLECTIVE BEHAVIOUR
Ant Colony Optimization with Parallel Subsolutions Heuristic

Oleg Kovářík, Miroslav Skrbek
Czech Technical University in Prague, Faculty of Electrical Engineering,
Department of Computer Science and Engineering
Karolovo náměstí 13,
121 35, Prague, Czech Republic
email: kovarol@fel.cvut.cz, skrbek@fel.cvut.cz

KEYWORDS
Ant Colony Optimization, Travelling Salesman Problem, Discrete Optimization

ABSTRACT

For solving $\mathcal{NP}$-complete Travelling Salesman Problem (TSP) by combining solutions of its subproblems, we designed algorithm inspired by Ant Colony Optimization (ACO) modified by adding parallel subsolutions heuristic. ACO can be parallelized simply by simultaneous execution of the algorithm with eventual exchange of the best solutions between all computational units. This approach requires access to a whole state matrix (which is of size $O(n^2)$) for each of them. It can limit the size of solvable problems on special architectures with different available memory capacities such as Cell Broadband Engine Architecture (CBEA). The presented algorithm keeps only one pheromone matrix in the memory of a main unit. The matrix is updated by subsolutions computed by ACO on other units in parallel. We show that this approach performs significantly better than greedy algorithm, even though it generates the whole solution from solutions of subproblems.

INTRODUCTION

The aim of this work is to show that a fairly good solution of the hard discrete optimization problem can be found by combining solutions of subproblems. Good tool for this task is a pheromone matrix used in algorithms inspired by ant behaviour. Information about subsolutions can be stored in this matrix and thus create probabilistic mixture of the good parts of the solution. By simulation of the pheromone driven stochastic movement of ants, we can generate the whole problem solution influenced by that mixture. It is interesting to find out, how close to the optimum we can approach by combining solutions of subproblems.

Before presenting our algorithm, we provide a quick overview of the solved discrete optimization problem and the original form of the ACO algorithm. It should be noted that presented algorithm can be used to solve other optimization problems in a way similar to the original one. Our algorithm uses no special features of the hardware platform with an exception of multiple computational units and fast interconnection grid.

Travelling Salesman Problem

Travelling Salesman Problem (TSP) is an $\mathcal{NP}$-complete discrete optimization problem. It is often used as a benchmarking problem for discrete optimization algorithms. The problem is to find a closed path for a given set of cities or a matrix of their distances. This path must contain each of $n$ cities exactly ones. The number of these paths grows very fast with the number of cities and the goal is to find the path with the minimal length. In this article, we solve a symmetric form of the TSP which means that the distance between two cities is the same in both directions. For more information on TSP see (Applegate et al. (2007)).

The TSP problem can be solved by using algorithms of the different complexity including greedy and k-opt heuristics, very successful V-opt method (Lin-Kernighan-Johnson), nature inspired Genetic Algorithms, Simulated Annealing, Tabu Search and others. Last but not least, MMAS algorithm derived from Ant Colony Optimization metaheuristic proved to be successful in solving TSP (Stützle and Hoos (2000)). Today’s most powerful methods for solving TSP use special properties of the Euclidean space. That is why they perform far better than the more general ones. An example of such specialized algorithm is Concorde (Applegate et al. (2007)).

Ant Colony Optimization

ACO is a metaheuristic approach to solve hard combinatorial optimization problems (Dorigo and Stützle (2002)). It is inspired by the indirect communication in real ant colonies that is mediated by laying and following the pheromone trails. When searching for food, ants use a pheromone trail to mark their paths between the food and the nest. Other ants are more likely to follow the trails with greater amount of pheromone and if they find enough food, these trails are reinforced. Random deviations from their path also make it possible to
gradually shorten the length of the path. Pheromone
evolution allows to forget less convenient paths.
ACO metaheuristic uses artificial ants and pheromone
to imitate a food searching behaviour of real ants. Algo-
rithms built on this metaheuristic are stochastic as the
ant movement is lead by probability function. They are
iterative since they repeatedly simulate ant movements
attended by pheromone laying and pheromone evap-
orating process:

procedure ACO_Metaheuristic() {
    while (termination_criterion_not_satisfied) {
        ants_generation_and_activity()
        pheromone_evaporation()
        daemon_actions() // optional
    }
}

The probability of moving from city $i$ to allowed (unvis-
itied) city $j$ is given by

$$
p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{j \in \text{allowed}} [\tau_{ij}(t)]^{\alpha} [\eta_{ij}]^{\beta}} & \text{if } j \in \text{allowed} \\
0 & \text{otherwise} \end{cases}
$$

(1)

where $\tau_{ij}(t)$ is the amount of pheromone between city
$i$ and $j$, $\eta_{ij} = 1/d_{ij}$ called visibility is computed from
distance $d_{ij}$ between cities, and $\alpha$ and $\beta$ are parameters
which allow the user to control the importance of the
pheromone and visibility.

After ants constructed their paths, pheromone between
every two cities is evaporated and then reinforced:

$$
\tau_{ij}(t + 1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t)
$$

(2)

where $\rho$ is evaporation factor, and the amount of the
pheromone to be added $\Delta \tau_{ij}(t)$ is a sum of amounts
from all $m$ ants:

$$
\Delta \tau_{ij}(t) = \sum_{k=1}^{m} \Delta \tau_{ij}^k(t)
$$

(3)

for

$$
\Delta \tau_{ij}^k(t) = \begin{cases} Q/L^k(t) & \text{if } (i, j) \text{ is a member of the tour} \\
T^k(t) \text{ of the } k\text{-th ant} \end{cases}
$$

(4)

where $Q$ is a constant value and $L^k(t)$ is the tour length
of the $k$-th ant.

In this work we use MAMAS algorithm (derived from
ACO metaheuristic) which limits the possible amounts
of pheromone to the interval $< \tau_{\text{min}}, \tau_{\text{max}} >$ to keep the
probability of choosing each edge greater than zero. This
can enable an escape from a local optima in the later

phases of the algorithm run. The second modification is
that each ant chooses to go to the nearest unvisited city
with a probability $q$ instead of using equation 1 (greedy
behaviour).

Figure 1: An example of pheromone layout during the
ACO algorithm run. Amount of pheromone between
cities is indicated by line thickness. More pheromone
means bigger probability of choosing this edge by the
travelling ants.

Parallelization

ACO can be easily parallelized thanks to its stochastic
nature. The problem can be solved on separate
computational units by simulation of multiple colonies.
Each colony solves the problem and the best solution is
taken as the result. We can also exchange the best solu-
tions between colonies using different strategies (Man-
frin et al. (2006)).

ACO WITH SUBSOLUTIONS HEURISTIC

ACO uses pheromone matrix to store information about
previous solution candidates. This pheromone is than
used as a heuristic information by an ant while choosing
next city to visit. Main idea of the proposed algorithm is
to use the information about solutions of subproblems
as a heuristic. This enables us to employ computational
nodes with low memory capacities, at the cost of lower
solution quality. Construction of the final solution is
guided by a probabilistic mixture of paths gained as
solutions from subproblems. This mixture is stored in
the main pheromone matrix.

Instead of updating pheromone matrix by a group of
ants, only one ant exists on the central unit. This ant
generates short paths which define subproblems; city
coordinates for these subproblems are distributed to all
Pseudocodes of the algorithm

Pseudocode for the main unit which distributes work and updates main pheromone matrix is:

```python
while (terminationCriterionNotSatisfied) {
    unit = wait_for_message()
    msg = get_message(unit)

    if (msg == READY) {
        size = choose_random_size()
        // ant movement:
        subp = generate_subproblem(size)
        transfer_coordinates(subp, unit)
    } else if (msg == DONE) {
        path = transfer_subsolution(unit)
        update_pheromone(path)
        if (num_of_updates > N) {
            evaporate_pheromone()
        }
    }

    if (num_of_evaporations > TRESHOLD) {
        // ant movement:
        generate_solution_candidate()
    }
}
```

Pseudocode for units used for solving subproblems is:

```python
while (termination_signal_not_received) {
    send_message(READY)
    wait_for_coordinates()
    receive_coordinates()
    solve_subproblem() // using ACO (MMAS)
    transfer_solution()
    send_message(DONE)
}
```

EXPERIMENTS

For experimental evaluation we use several instances of the symmetric Travelling Salesman Problem from the widely used problems library TSPLIB (Reinelt (1991)). It was used in many studies on TSP and contains both real-world and artificial datasets.

In the beginning optimal parameters had to be found. Dataset st70 from TSPLIB was used for experiments on parameters settings. Random experiments on other datasets seemed to give similar results. $\alpha$, $\beta$ and $q$ parameters were set according to recommendations for MMAS algorithm. For the main unit, they were set to more "greedy" behaviour (bigger $\beta$ and $q$ parameter). After few random experiments, evaporation factor $\rho$ was set to 0.01 and $\beta_{main}$ was set to 10.0 to achieve algorithm convergence in a reasonable time. After that,
Figure 3: A comparison of the average path lengths for various values of the pheromone factor parameter $v$. Shorter path means better solution.

pheromone update factor $v$ was chosen as the most sensitive of the remaining parameters as it has the greatest impact on convergence to near-optimal solutions. Series of experiments was conducted and the results can be seen in figure 3.

Optimal value for $q_{main}$ was then determined in experiments (see figure 4).

Figure 4: A comparison of the average path lengths for various values of the greedy move probability parameter. Shorter path means better solution.

Resulting configuration can be seen in table 1. It was used for the main part of experiments. Interesting is $\beta_{main}$ parameter which causes stronger “greedy” behaviour of the ant on the main unit.

Typical algorithm run is presented in the figure 5. It shows how the algorithm converges to the value between optimum and greedy solution.

As a subproblems solution we can use opened paths or closed paths. A part of experiments was the inquiry which of these settings is better. As we expected, better performance was achieved when open paths were used. Sizes of the subproblems were random ranging from 5 to 14. Solving larger subsolutions was too slow to have some major effect on solution quality.

As a termination criterion for the algorithm, the number of evaporation cycles without improvement greater than $n \times 300$ was used, which seemed reasonably large for examined TSP instances. For larger instances, adjustments in parameters may be required.

For our experiments we used Cell Broadband Engine Architecture (CBEA) which is a heterogeneous multi-core processor with one main unit and six other units (in the version used in PlayStation 3) with its own small “Local Storages”, capable of holding subproblems of the solved problem. Transfers of data between main memory and “Local Storages” are performed by DMA.

**RESULTS**

We evaluated our algorithm on six datasets from TSPLIB and compared the results with optimal and greedy solution. Greedy solution is calculated for each city and the smallest from resulting path lengths is used for comparison. Results of these experiments are summarized in table 2. Values for ACO with subsolution heuristic are average lengths from 10 runs of the algorithm. Our algorithm produces up to 10% longer paths.

Table 1: The best algorithm parameters found. These parameters play a part in equations 1 and 5. Parameters with index “main” are used on the main unit.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>0.01</td>
</tr>
<tr>
<td>$v$</td>
<td>0.02</td>
</tr>
<tr>
<td>$q_{main}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$q$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\beta_{main}$</td>
<td>10.0</td>
</tr>
<tr>
<td>$\beta$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Figure 5: An example of algorithm run together with greedy and optimal solution.
than the optimal ones, and the solution lies between optimum and greedy value. This result can be improved by usual 2-OPT local search, but this improvement was not used to prevent a distortion of results.

Table 2: Results for six datasets from TSPLIB. ACO with subsolutions is compared with the greedy solution and the optimal solution.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>eil51</th>
<th>pr76</th>
<th>rat99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy solution</td>
<td>482</td>
<td>130921</td>
<td>1437</td>
</tr>
<tr>
<td>Optimal solution</td>
<td>426</td>
<td>108159</td>
<td>1211</td>
</tr>
<tr>
<td>ACO+subsolutions</td>
<td>449</td>
<td>118644</td>
<td>1318</td>
</tr>
<tr>
<td>% above optimum</td>
<td>5.4</td>
<td>9.7</td>
<td>8.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>kroA100</th>
<th>lin105</th>
<th>bier127</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy solution</td>
<td>24698</td>
<td>16935</td>
<td>128726</td>
</tr>
<tr>
<td>Optimal solution</td>
<td>21282</td>
<td>14379</td>
<td>118282</td>
</tr>
<tr>
<td>ACO+subsolutions</td>
<td>22922</td>
<td>15262</td>
<td>125809</td>
</tr>
<tr>
<td>% above optimum</td>
<td>7.7</td>
<td>6.1</td>
<td>6.4</td>
</tr>
</tbody>
</table>

CONCLUSION

We designed algorithm for solving \(\mathcal{NP}\)-complete Travelling Salesman Problem by combining solutions of its subproblems. Algorithm is a modification of the ACO metaheuristic and uses its pheromone matrix data structure to store heuristic information about subsolutions. Parts of the main problem are solved on the computational units with low memory capacities. Whole solution state (pheromone matrix) is kept only in a memory of the main unit and is continually updated by solutions of subproblems.

Experiments on six datasets from TSPLIB show that this algorithm gives better results than the best greedy solution. This is possible thanks to probabilistic solution construction guided by two heuristics: distance heuristic as in standard ACO and subsolutions heuristic in the form of the pheromone matrix. Path lengths are from 5% to 10% above optimal value. No local optimization is performed to avoid false positive interpretation. Updates of the pheromone by the solutions generated on the main unit are avoided for the same reason. It would be interesting to combine these two approaches to see if such combination can result in a faster convergence of the ACO algorithm or better results.

An advantage of this approach is that the main unit only distributes work, performs updates from collected results and generates candidate solutions from time to time. This reduces main unit load and enables to utilize units with low memory capacities at the expense of lower solution quality than standard ACO. An interesting property of proposed algorithm is a construction of \(\mathcal{NP}\)-complete problem solution from its subsolutions in the probabilistic way. The potential of this approach should be examined further.

We used heterogeneous multi-core Cell Broadband Engine Architecture as a hardware platform for the experiments because it matches the architecture of the algorithm and its high performance enables us to conduct experiments in the reasonable time. To fully exploit possibilities of this platform, it is possible to replace the ACO algorithm on the satellite units with another optimization algorithm, which can make greater use of the vector and multimedia processing possibilities.

ACKNOWLEDGMENT

This work is supported by the internal grant of Czech Technical University in Prague, IG: CTU0807213 - “New modifications to ant algorithms” and the research program “Transdisciplinary Research in the Area of Biomedical Engineering II” (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

REFERENCES


SIMULATION OF IMMUNE SYSTEM RESPONSE TO BACTERIAL CHALLENGE

Nuno Fachada1,2*, Vitor V. Lopes1,2, Agostinho Rosa1

1 LaSEEB - Instituto de Sistemas e Robotica,
Torre Norte 6.21, Av. Rovisco Pais 1, 1049-001 Lisboa
2 INETI - Instituto Nacional de Engenharia, Tecnologia e Inovação
Estrada do Paço do Lumiar, 22, 1649-038 Lisboa

KEYWORDS
Modeling, Simulation, Agents, Immune System

ABSTRACT

Immune system (IS) simulations have several applications, such as biological theory testing or as a complement in the development of improved drugs. This paper presents an agent based approach to simulate the IS response to bacterial infection challenge. The agent simulator is implemented in a discrete time and two-dimensional space, and composed by two layers: a) a specialized cellular automata responsible for substance diffusion and reactions; and b) the layer where agents move, act and interact. The IS model focuses upon low level cellular receptor interactions, receptor diversity and genetic-ruled agents, aiming to observe and study the resultant emergent behavior. The model reproduces the following IS behavioral characteristics: specificity and specialization, immune memory and vaccine immunization.

INTRODUCTION

Experimental immunology research is a difficult and expensive research field, where the systematic clinical trials of new drugs follow a complex and strict protocol (Emerson and Rossi 2007). As such, computational models of the immune system (IS) can be a valuable tool to understand the effects of a new drug, as well as to test or validate immunologic theories. In agent-based modeling (ABM), a system is modeled as a group of independent decision-making agents that evaluate its current situation and make decisions on the basis of a rule set (Chen et al. 2004). Cellular automata (CA) (Wolfram 2002) are the simplest form of ABM, and are based on an environment with non-moving agents, discrete in space and time. Agents can range from simple propositional logic based agents (Remondino 2003) to learning agents (Bonabeau 2002) (e.g., using neural networks or evolutionary algorithms).

Several ABM models to perform IS simulation already exist, each presenting a different focus, approach and features. ImmSim, one of the most referenced and peer reviewed IS simulators, is based on a CA with probabilistic rules (Kohler et al. 2000), presenting concepts later used in other models, such as entities moving from site to site. The fundamental concepts of this model were explored in 1992 (Celada and Seiden 1992). The AbAIS (Agent-based Artificial Immune System) introduced a hybrid approach supporting the evolution of an heterogeneous population of genetic-ruled agents over a CA environment (Grilo et al. 2001). Simmune brought the modeling focus on low level molecular interactions (Meier-Schellersheim and Mack 1999), although with limited results regarding IS simulation. Event driven IS modeling was introduced in CAFISS, a platform which used multithreaded asynchronous updating of the simulation (Tay and Jhavar 2005), where each IS cell instance runs in its own thread; although realistic, this is a computationally expensive approach. The concern on specialized engines to manage physical and chemical interactions was underlined by the Sentinel platform, used in the evaluation of several immunological memory theories (Robbins and Garrett 2005).

In this paper we discuss an agent-based model of the IS and bacteria, and present several simulations of the IS response to bacterial attack under different circumstances. The model is developed using LAIS, a framework for simulation of biological systems in general and the IS in particular (Fachada 2008), which gathers and improves on important features of previous models, offering a versatile and accessible modeling approach. Simulations concern immune memory, specificity and specialization; a vaccine simulation is also presented.

THE IMMUNE SYSTEM

The exact function of the IS is still a source of active debate, but it can be stated that its physiologic function is to protect individuals against infections (Abbas and Lichtman 2006) caused by pathogenic agents. At the same time the IS must distinguish self from non-self, in order to avoid self inflicted damage. Autoimmune diseases are the consequence of failure to perform such distinction. The IS is gifted with learning and memory features: it remembers previous challenges with specific pathogens, and deals with them more effectively in subsequent encounters. The defense mechanism of an indi-

*Corresponding author.
vidual consists of innate and adaptive immunity, which work together to provide protection against infections. Innate immunity is the first line of defense against infections; its performance does not depend on prior contact with potential threats. Innate immunity cells, such as macrophages, recognize generic pathogen-associated molecular patterns in the surface of microbes, destroying them via phagocytosis (i.e., by engulfing them) (Abbas and Lichtman 2006).

Microbial adversaries can rapidly evolve strategies to evade innate immunity mechanisms. Adaptive immunity is the evolutionary answer of vertebrate animals, allowing the body to adapt to first time invasions, remembering and handling them more effectively in the future. Lymphocytes are adaptive immunity agents which can challenge particular invaders through the recognition of the unique receptors they express, known as antigens. There are two main types of lymphocytes, which differ in function and type of antigen receptor: B cells and T cells.

The B cell produces antibody molecules complementary to a given antigen in its native form; it plays a central role in humoral immunity, the protection against extracellular microbes. When a macrophage detects a microorganism covered (opsonized) with antibody, the probability of successful phagocytosis increases substantially. The B cell is activated when its receptor (BCR), a superficial antibody, binds specific antigen. The T cell is the main actor in cell-mediated immunity (CMI), which provides protection against intracellular microbes. T cells are subdivided in Th (helper) cells, which assist macrophages and B cells, and Tc (cytotoxic) cells, which kill infected cells. Th cells may also help the activation of Tc lymphocytes. The T cell receptor (TCR) is more complex, and will not bind to native antigen; instead, it binds a complex formed by an MHC molecule and an antigen derived peptide. MHC (Major Histocompatibility Complex) is a genetic receptor of body cells, involved in antigen presentation to T cells; MHC class I is present in all nucleated cells and is recognized by Tc cells; MHC class II exists on antigen-presenting cells (mainly B cells, macrophages, dendritic cells), and is recognized by Th cells (Roitt and Delves 2001). Cells that present antigenic peptides to Th cells via MHC class II, such as macrophages and B cells, are known as Antigen Presenting Cells (APC). Macrophages process antigen for presentation after microbe phagocytosis, while B cells do the same after engulfing BCRs binding antigen.

When a B cell presents antigen to a Th cell, the latter is stimulated to secrete cytokines (mediators of immune and inflammatory reactions (Abbas and Lichtman 2006)), which in turn increase B cell proliferation and differentiation. B cells either become plasma cells, which secrete antibody, or long-lived memory B cells, which allow a more effective response in future challenges by the same microorganism. After a few days, some of the antigen activated B cells undergo a process called somatic hypermutation, which consists of high-frequency mutations in antibody specificity; B cells producing higher affinity antibodies after mutation have an increased chance of survival, leading to affinity maturation of the humoral immune response. When a Th cell recognizes the antigenic peptide + MHC class II complex on the surface of a macrophage, it releases IFN-γ, a cytokine which helps the macrophage destroy phagocytosed, but still living microbes. Several bacteria, such as Listeria, Mycobacterium tuberculosis or M. leprae, survive inside macrophages, requiring external macrophage activation by Th cells in order to be properly eliminated. These are important aspects of IS dynamics and adversarial strategies, and illustrate the variety of ways in which different components interact in order achieve their goal. However, this introduction doesn’t even begin to reflect the true complexity of what is at stake; it serves only to contextualize the reader and to establish an underlying natural agent-based structure, further justifying the use of agent-based approaches for modelling the IS.

THE LAIS SIMULATION FRAMEWORK

The LAIS framework is a multi-threaded agent-based simulation platform, offering a set of tools for the simulation of biological systems. The platform is implemented in Java and makes use of following open source libraries: a) the Repast Agent Simulation Toolkit (North et al. 2006) classes that provide or simplify spatial organization and visualization, event scheduling and simulation output (e.g., charts, CSV files, movies); and b) the Simple XML serialization library1 that provides simple class development and instantiation using XML. The platform will be available on Sourceforge early 2009. The two main actors in the LAIS framework are the substances and the agents. The simulator is organized in two layers: a) a specialized cellular automaton (CA) responsible for substance diffusion, reaction and degradation; and b) the agent layer where the agents move and act. The communication between these layers occurs when agents produce or consume substances, or when an agent action depends on the underlying substances. Current implementation restricts the simulation to discrete time and two-dimensional space.

Substances

Substances are uniquely identified by a 64 bit string, allowing a repertoire of $2^{64} \approx 10^{19}$ different substances. In the model specification it is possible to attribute specific biological functions to different bit substrings. The biological affinity between substances primarily depends on the existence of complementary zones, i.e., regions where

1http://simple.sourceforge.net/
the biological substances can “fit” with each other. To mimic the IS, the bit string of substances that model IS antibodies are composed by: a) a constant region responsible for secondary functions such as macrophage binding or complement fixation, and b) a variable region which is used to determine the binding affinity with the antigen. The biological affinity is implemented by the Hamming distance between two substance bitstrings (Celada and Seiden 1992).

LAIS represents the substances as real valued concentrations, allowing to: a) model diffusion and reaction phenomena in the CA layer; and b) simulate the substances present on the agent surface, in the agent layer. Antigens are modeled as substances and thus differentiated from pathogenic agents themselves. New substances can be dynamically created during simulation as the result of: a) different substances produced by mutation of cloning agents; and, b) substance merging. Merging can be either affinity dependent, such as in the antigen-antibody complex formation or independent, such as in the case of the complex formed by MHC and the antigen peptide.

The simulator offers the possibility to group the substances into families in order to: a) simplify the process of tracking substances with similar functions, e.g. in B cell response, where a multitude of different antibodies are temporarily produced; and, b) allow the definition of substance merging rules affecting specific families. In the latter case, model specification is considerably facilitated and the substance merging simulation becomes computationally feasible.

Agents

Agents have a set of conditional rules which evaluate state, superficial substance concentration and the local CA cell, analyzing local substance concentration, as well as substances displayed by other agents. These rules are grouped in lists of rules; each list of rules is associated to a list of actions. A “rule list - action list” mapping is called a “gene”. In order to perform the actions in a list, all the rules in the associated gene rule list must yield true. Fig. 1 shows the schematics of a LAIS agent. Rules and actions are hard-coded Java classes, but accept instantiation parameters, making them flexible. The grouping of rules and actions with different instantiation parameters permits a vast range of behaviors. If a particular behavior cannot be achieved using available rules and actions, it is relatively simple to code additional ones, following specific interfaces. The agent set of genes (each one being a “rule list - action list” mapping) can be referred as the agent’s genotype. Evolution takes place when an agent creates another agent, either by a cloning process (e.g., cellular division) or by producing a different type of agent (e.g., an infected immune cell producing viruses). In such cases, rules and actions are also cloned. These have a mutation parameter which can modify referenced substances. Agent movement is controlled by these rules; movement can be random, inertial (higher probability of moving forward) or substance dependent (simulating chemotaxis, cell movement directed by a chemical concentration gradient (Abbas and Lichtman 2006)). LAIS supports the exchange of genetic code (at gene level), a feature introduced in AbAIS, allowing models to represent realistic evolving systems.

MODELING CONSIDERATIONS

One of the most complex challenges when developing a IS model is to find a balance between scale, granularity and computational feasibility. Features that are included in the model should not only be theoretically and experimentally sound, but also relevant for in the context of the simulations to perform. Knowledge gaps, incomplete data and excluded system features imply that models are incomplete, always abstract to some point. However, an incomplete model can still do a good job of simulating reality. Biological systems can also work without various parts; they are robust, having redundant features and components with overlapping functions. As such, if a model captures the principal components of a biological system, there is no reason why it cannot yield realistic simulations (Cohen 2007).

Having the previous paragraph into consideration, four types of entities were modeled in order to perform a simulation of immune response to bacterial challenge: APCs, B cells, Th cells and a phagocytosis resistant bacteria agent. The most relevant soluble substances for this experiment consisted in three cytokines, a single bacterial antigen, and a variety of antibodies and antigen + antibody complex, not known at the beginning of the simulation, as its production is a consequence of the immune response. Other substances, such as MHC Class II, are only present in the agent surface, but are of critical importance in the overall simulation. The behavior of IS agent models follows the description discussed in
the IMMUNE SYSTEM section, while the model bacteria consists of an agent who’s replication rate is higher than the death rate, so if left alone in the simulation environment would grow indefinitely; naturally, limited resources would stop this from occurring in reality. Detailed model implementation can be found in (Fachada 2008).

EXPERIMENTS AND RESULTS

Immune memory

The establishment of memory against previous infections is one of the most important characteristics of the adaptive immune system. This property is responsible for enhanced immune responses to recurrent infections (Abbas and Lichtman 2006). Bacteria are inserted for the first time in the simulation environment at tick 95 (fig. 2a). The immune response ensues, with APCs performing phagocytosis. APCs are unable to kill the ingested bacteria, but the release of IL-12 cytokine and presentation of MHC Class II + antigen complex activates specific Th cells. These release IL-2, which induces self-proliferation and proliferation of B cells, a process called clonal expansion (i.e., the multiplication of cells specific for the invader). Th cells continue to proliferate, and become effector cells, i.e., cells which produce IFN-γ. This cytokine will in turn help the APCs “kill” the phagocytosed bacteria. Antigen specific B cells also take part in the initial activation of Th cells, as they also express MHC Class II + antigen complex, after engulfing soluble antigen. However, they only become activated after IL-2 signaling. Some of the activated B cells become antibody producing plasma cells, while others will go into somatic hypermutation state and others become memory. The antibodies produced by plasma cells act in two fronts by a) opsonizing the bacteria, helping APCs perform phagocytosis, and b) directly neutralizing bacteria when concentration is high enough. B cells in somatic hypermutation undergo affinity maturation, in which only high affinity clones are able to survive; the surviving clones then become antigen specific activated, plasma and memory cells. When the concerted response is in place, bacteria starts to be removed from the environment, lasting 51 ticks from initial insertion to full neutralization. When all bacteria are eliminated, the immune response is hampered by the negative feedback rules in the agents. All traces of the first bacterial challenge, except for the long-lived memory cells, are cleared by tick 500 (fig. 2). At tick 533, a new dose of bacteria is introduced in the simulation environment (fig. 2a); it’s possible to observe that they resist less when compared to the primary attack, surviving 18 ticks. The secondary response had a shorter delay between antigen deployment and antibody production, and the production of antibody was higher, as can be observed in fig. 2b. These results compare favorably with literature descriptions (Abbas and Lichtman 2006, Roitt and Delves 2001).

Specificity and specialization

Like memory, specificity and specialization are two important properties of the adaptive immune system. Specificity is the ability to recognize and respond to a variety of microorganisms, while specialization refers to the fact that responses for distinct microbes are optimized for defense against these microbes (Abbas and Lichtman 2006). This experiment aims to demonstrate these two characteristics. In order to verify specificity and specialization in the model, two types of bacteria, A and B, are introduced at simulation ticks 50 and 800, respectively (fig. 3a). The bacteria are distinguished only by their superficial antigen. The response to the first challenge leads to the creation of specific memory against bacteria A. When the secondary challenge occurs, there is no evident improvement in the quality of the secondary response (fig. 3b), with bacteria B surviving slightly longer (fig. 3a). This occurs because memory cells created during the
first challenge are specific for bacteria A, not recognizing bacteria B during the secondary response. Comparing the antibody response (fig. 3b) with results from literature (Abbas and Lichtman 2006, Roitt and Delves 2001), it is possible to conclude that the model yields the expected results.

![Number of bacteria.](image1)

Figure 3: Specific memory created for bacteria A does not recognize bacteria B.

**Vaccine simulation**

Vaccination uses the memory property of the immune system to achieve a state of immunization against a given pathogen. The first stages of vaccine testing constitute a perfect opportunity for the use of simulators, which can provide preliminary indications on the effectiveness and safety of new drugs. The use of killed organisms as a vaccine is one of the most common, although with limited effectiveness (Roitt and Delves 2001). In this experiment, a virtual vaccine based on dead (non-replicating) bacteria is used to immunize the host, and its effectiveness measured (fig. 4).

The vaccine, composed of 50 non-replicating bacteria, is introduced at tick 50 (fig. 4a) (in the figure the number does not reach 50 due to the simulation mechanism: first the vaccine is deployed, then cells perform their steps - some vaccine disappears - and only then are the charts updated). The immune response is eventually mounted, removing the vaccine from the system and creating memory cells in the process. At tick 500, 25 units of “live” replicating bacteria are deployed in the simulation environment (fig. 4a). The IS recognizes the invader and the response is almost immediate when compared with the initial injection of vaccine (fig. 4b). The immune response removes the bacteria without difficulty, asserting the effect of vaccination using “dead” microorganisms in the presented model.

![Vaccine composed of attenuated bacteria causes a state of immunization.](image2)

Figure 4: Vaccine composed of attenuated bacteria causes a state of immunization.

**CONCLUSIONS**

This paper presents an agent based approach to simulate the immune system response to bacterial challenge. It aims to demonstrate LAIS framework capabilities for this type of simulations. Three simulation scenarios were designed to test the implemented model for immune memory, vaccine immunization and the specificity and specialization behavioral characteristics present in the IS. Results show that: a) LAIS framework archi-
tecture provides all the necessary flexibility to specify the IS model in a biological meaningful way; and b) the implemented IS model can reflect all these three characteristics and, thus, be further calibrated and validated against experimental data.

ACKNOWLEDGMENTS

This work was supported by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS_Colheitamento Program that includes FEDER funds.

REFERENCES


URBAN CULTURAL DYNAMICS MODELLING USING SWARM INTELLIGENCE AND GEOGRAPHICAL INFORMATION SYSTEMS

Rawan Ghnemat(1), Françoise Lucchini(2), Cyrille Bertelle(1)

(1)LITIS (EA 4108) - 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)IDEES - MTG (UMR 6228) - University of Rouen, rue Lavoisier, 76821 Mont Saint Aignan, France
email: francoise.lucchini@univ-rouen.fr

KEYWORDS
swarm intelligence, complex systems, urban dynamics, geographical information systems, cultural equipment, self-organization, ant systems, spatial organization

ABSTRACT

This paper deals with the understanding of collective cultural shape dynamics within urban area. The cultural facilities inside the city are various and follow complex spatial mecanisms. Both geographical aspects and social factors are major in these dynamics, leading to specific equipment. The complexity of these phenomena which are the basis of the cultural development, needs some specific modelling technics. In a first step, we propose some analysis and description of the cultural activity distribution based on a specific geographical information system (GIS). A case study is developed for the french city of Rouen in Normandy. In a second step, and respecting the complexity of the studied systems, we propose to use swarm intelligence models over this GIS to model the urban cultural dynamics. The goal is a better understanding of the relevent interactions within multi-criteria interactions.

HOW MODELLING THE DIFFUSION OF CULTURAL INFRASTRUCTURES WITHIN THE CONTEXT OF URBAN DYNAMICS COMPLEXITY

Urban dynamics are the result of complex systems de
velopment where a great number of entities and factors interact. The bottom-up class of modeling consists to define the city as a collection of individual-based description, behavioral rules-based description and interaction systems. From this constructive approach, we want to obtain an emergent description of the whole system or of some sub-systems included in a hierarchical process. Two complementary methodologies can be used for that and we detail them in the following paragraphs.

The first methodology consists to generate a simulation where all the components, behaviors and rules-based, interact over a environment, perceiving and acting on it. The environment evolving is the support of emergent properties. The cellular automata modeling deals with this kind of simulations. The basic definition of cellular automata for urban or regional modeling, for instance, consists in the decomposition of the city, region or any geographical area in a lattice of cells. Each cell is in some state which belongs to a finite set S. At each time step, the cells change its own state according to some transition rules based on its previous state and its neighbor cells. Many works based on cellular automata, have been developed for geographical systems and urban dynamics (2; 9; 15). To deal with some urban dynamics, we need an important extension to cellular automata where we have to represent individual moving, for instance. The mixing of spatial data and cellular automata with autonomous entities, like agents, is here needed (7).

The second methodology to deal with emergent description in micro-modelling, consists to complete the previous approach based on simulation, by introducing some computational processes which are able to detect emergent systems or organizations. The final goal of this method is then to be able to re-introduce these emergent systems or organizations inside the simulation and manage their evolutions and their interactions with the components of the system. The re-integration of the emergent systems, during the simulation, can be explicitly expressed like in the multiscale fluid flow simulation proposed by P. Tranouez (25) or it can be implicitly expressed using an adaptive process as we will describe in the following.

Cultural equipment development within urban dynamics must be understanding through two aspects: the specific mechanisms which control the creation of cultural infrastructures and the use and attraction phenomena of
these cultural equipments to follow their development. Emergent systems and organizations simulation must be implement to analysis the cultural dynamics and we will propose swarm intelligence technologies to model these phenomena.

**ANALYSIS OF CULTURAL ACTIVITY DEVELOPMENT BASED ON GIS - A CASE STUDY FOR THE FRENCH URBAN AREA OF ROUEN**

Our goal is to study the diffusion of cultural facilities and propagation during time, inside the urban area of Rouen. A contructive approach is proposed in order to understand the spatial dynamics of cultural infrastructures development. We want to better understand what are the relevant interactions within heterogeneous phenomena.

**Data Support**

We design the cultural mechanism in a social and territorial context. The cultural activities and equipment are particularly all services present in French city (20), where the majority of the population is located in, considering the general urbanization of the French Society. Like others French cities, Rouen proposes a large diversity of cultural activities in the hopes of educating, entertaining and satisfying inhabitants, but also for self-promotion.

The data from the city of Rouen concern urban environment and public or private cultural equipments within wide vision of culture. There are with academic sites (museums, operas, theatres, libraries, ...) and popular ones (musical pubs, concert places, festivals, cinemas, ...). These data have been obtained from the agglomeration population composed of 400,000 inhabitants. We plan to study the various logic of geographical repartition and the motivation to create and build cultural equipment, including strategies for municipalities or commercial exploitations.

At this local level we wish understand two mechanisms. The first one is the conditions that conduct people or private and public institutions to create new cultural structures and how the new locations are chosen. Second one is to better know the circulation of the inhabitants during time in these cultural places which have different levels of attraction. Because, in addition to the problem of geographical distances, other parameters play an equal role in accessing culture: social, educational and economic selection constitute another restraint for the cultural practices of citizens (5).

**First Graphical Analysis using GIS**

Urban environment is based on land use of Rouen from topographical maps from IGN. Various GIS have been developed from these maps, mixing communication networks and cultural edifices (see on figure 1) or mixing land use and cultural equipments (see the figure 2). The mixing with the municipality administration and service attractivity is also plan to be studied.

![Figure 1: Communication networks and cultural equipment development mixing](image)

**Processes Analysis and Formalization**

Cultural equipment repartition within city does not follow stochastic distribution (26). This implies that the geography of these activities call on other dynamics than that of chance. This repartition is very different from sports equipment development, for example, as we can see on figure 3. The sports development follow the public school development that the cultural equipment does not follow.

These observations lead us to better understand what are the mechanisms involved in cultural phenomena. We have to study first the specific spatial constraints: river separation for the city of Rouen, industrial and housing proximity and municipalities service location. Spatial constraints are so a major aspect within the development dynamics, including heterogeneous factors. To model such complexity, we propose a swarm intelligence method base on ant systems as detailed in the next section.
MIXING GIS AND SWARM INTELLIGENCE FOR THE DYNAMICS SIMULATION

We will describe in this section, the general algorithm which is proposed to model emergent spatial organizations within cultural dynamics. This algorithm is based on the ant clustering. We introduce pheromone template to spatially control the clustering from local attraction.

This method is a decentralized approach which allows to combine multi-center and multi-criteria problems.

Ant clustering

Ant clustering algorithms are inspired by the corporoses or larvea classification and aggregation that the ants colony are able to do in the real life. The ants are moving inside a closed area and are able to move some material which are randomly put on this area. After a while, and without any kind of centralized coordination, the ants success to create some material clusters.

The algorithm is based on the following and very simple behavioral rules that each ant implements:

- When an ant is moving without carrying yet material and find some material, the ant will take the material respecting the probability number:
  \[ P_p = \left( \frac{k_1}{k_1 + f} \right)^2 \]  
  (1)
  where \( f \) is the material density that the ant perceives locally around itself and \( k_1 \) is the threshold. It is easy to check that if \( f << k_1 \) then \( P_p \) is near the value 1 and if \( f >> k_1 \) then \( P_p \) is near the value 0.
- When an ant is moving when carrying some material, the probability to deposite it is computed by:
  \[ P_d = \left( \frac{f}{k_2 + f} \right)^2 \]  
  (2)
where \( f \) is still the material density that the ant perceives locally around itself and \( k_2 \) is another threshold. It is easy to check that if \( f << k_2 \) then \( P_d \) is near the value 0 and if \( f >> k_2 \) then \( P_d \) is near the value 1.

Figure 4: Ant nest building with one center using RePast MAS paltform over OpenMap GIS

Spatial constraints using template

The ant clustering shows some spatial self-organizations but has the specificity to generate clusters at random places. According to the first random moves that the ants start to do in the beginning of the algorithm, some material will initiate aggregation and the clustering processus will complete this aggregation from these initial random first aggregations. To simulate some urban dynamics, we need to introduce specific location with respect to city center for example or cultural equipments. The clustering here will represent the people use of these centers or equipments and we need to introduce an attractive effect by using a pheromone template. This method follow the algorithm known as Ant Nest Building (4). In ant colonies, the center corresponds to the position of the queen which needs to build the nest and the ant colony moves around it to protect the nest by various material taken on the ground. The queen emits a pheromone which allows to attract the ants during their building. The ant has to deposite the material carried only if the pheromone quantity perceived belongs to a specific range. We use an attractive fonction called \( P_t \) corresponding to a pheromone template and represented by the part (a) of the figure 4.

Using this template fonction, we replace in the clustering algorithm, the two previous probabilities defined in equation (1) and equation (2) by

\[
P'_p = P_p(1 - P_t)
\]

\[
P'_d = P_dP_t
\]

In figure 4, we show an implementation of this algorithm using the multi-agent platform called Repast (21). The java version of this platform includes some packages allowing to interface with geographical database and geographical information systems (GIS). The graphical out-
put windows is made under OpenMap which is a GIS developed in Java. In figure, the materials moved by the ants are the small grey circles, the ant moving without material are the green circles, the ant carrying material are the red circles and the queen location is the yellow circle.

Multi-template modelling

The previous subsection describes one local attractive process characterized by the queen and its pheromone template emission. The advantage of this method is to be able to combine the solutions of multi-center and multi-criteria problems, using interactive processes, each one is represented by a queen and its pheromone template.

On the figure (5), we can see a simulation with two queens and two pheromone templates. It is possible also for each queen to emit many different kinds of pheromones: we called them colored pheromones. Each colored pheromone will attract only the ants associated to its color.

Application to cultural equipment dynamics

The multi-template modelling can be used to model cultural equipment dynamics as described in the figure (6). On this figure, we associate to each cultural center (cinema, theatre, ...) a queen. Each queen will emit many pheromone templates, each template is associated to a specific criterium (according to age, sex, ...). Initially, we put the material in the residential place. Each material has some characteristics, corresponding to the people living in this residential area. The simulation shows the self-organization process as the result of the set of the attractive effect of all the center and all the templates.

CONCLUSION AND PERSPECTIVES

We study in this paper the cultural equipment dynamics within urban area with a specific case study concerning the french city of Rouen. Geographical Information Systems have been developed from large database concerning Rouen Agglomeration and allow to highlight the complexity of this problem: the link between communication networks or land use with cultural development. To model the complexity of cultural center attractivity and use, we propose a swarm intelligence method using a bio-inspired model called ant nest building. This model allows to deal with spatial self-organization dynamics using multi-center and multi-criteria aspects. Implementation on Repast platform with GIS mixing is proposed as the basis or the development framework. Specific results on Rouen Agglomeration are still in progress and are expected to give relevant analysis to better understand these specific cultural dynamics.

REFERENCES


Figure 6: Cultural Equipment Dynamics Modelling


[28] Xiao, N. (2005) Geographic optimization using evolutionary algorithms In 8th International Conference on GeoComputation, University of Michigan, USA.
MANAGING SPATIAL SELF-ORGANIZATION VIA COLLECTIVE BEHAVIORS

Rawan Ghnemat(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: 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
swarm intelligence, complex systems, self-organization, ant systems, Schelling model, spatial organization

ABSTRACT

Spatial self-organizations appear in many natural and artificial systems. Spatial systems creation and development, called morphogenesis, is the subject of many research studies since many years (1). Fractal computation approach is, for example, one of the methods proposed to deal with such studies. But, even if this method is able to describe unlimited local formations on multi-scale descriptions, the formation process itself is described in a global way. The goal of this paper is to introduce the distributed and decentralized computing as a general methodology to propose emergent spatial formation, able to deal with local perturbations and with non homogeneous formation rules. We propose a study case on Schelling Model dealing with interacting population over an environment based on a regular grid.

SPATIAL MORPHOLOGY MODELLING ON THE EDGE OF COMPLEXITY

The study of spatial morphology is a major aspect of the understanding of many phenomena for natural or artificial systems. Living systems or social systems, for example, are systems where the spatial formation has a high meaning and modifies deeply by itself the system evolution. The system evolution leads to modify itself the spatial formations by feed-back processes.

Spatial morphology models can be classified by many criteria. Some of these models are static (finding the optimal shape of some problem) or dynamic (morphogenesis, for example). When the models involve dynamical processes, these dynamics can be expressed in a global way, like we do using partial differential equations: the objective of the system description consists in describing the different phenomena involved (diffusion, transport, ...).

Spatial morphology systems can be involved inside a multi-scale processus, giving some specific properties to this multi-scale formation, like the development of important exchange area. Fractal systems are well-known to model these multi-scale systems like fractal shape of plants.

Even if these fractal geometries are able to model multi-scale descriptions, they are generally completely deterministic and they are not suitable to describe geometrical evolutions or to integrate local disturbances. For this purpose, we need to change the model concept and go from global deterministic models to decentralized approaches where the whole system is only known (or emerge) by the interaction system of behavior population.

SCHELLING MODEL EXTENSION IN ORDER TO MODEL SELF-ORGANIZATION BY MEANING OF MULTI-CRITERIA SYSTEM SEPARATION

Thomas Schelling’s city segregation model illustrates how spatial organizations can emerge from local rules, concerning the spatial distribution of people which belong to different classes. In this model, people can move, depending on their own satisfaction to have neighbours of their own class. Based on this model, a city can be highly segregated even if people have only a mild preference for living among people similar to them.

In this model, each person is an agent placed on a 2D grid (in his original presentation, a chessboard was used by Thomas Schelling). Each case can be considered like a house where the agent lives. Each agent cares about the class of his immediate neighbours who are the occupants of the abutting squares of the chessboard. Each agent has a maximum of eight possible neighbours. He computes the rate of the neighbours of its own class
from its eight possible neighbours. Each agent has a tolerance rate determining whether he is happy or not at his current house location. If the rate of the neighbours of its own class is under this tolerance rate, he decides to move to live in another free place in the 2D grid.

![Figure 1: Initial situation](image1)

![Figure 1: Stable situation after 124 iterations](image2)

(a) Initial situation

(b) Stable situation after 124 iterations

Figure 1: Standard 2 populations segregation Schelling model on RePast

The exact degree of segregation which emerges in the city depends strongly on the specification of the agents’ tolerance rate. It is noticeable that, under some rule specifications, Schelling’s city can transit from a highly integrated state to a highly segregated state in response to a small local disturbance. We can observe some bifurcation phenomena which lead to chain reaction of displacements.

In figure 1, we show an implementation of this algorithm using the multi-agent platform called Repast (7). Here, we show the “classical” and original problem, modelling the segregation phenomenon with two population classes, described by red and blue squares. Both,

![Figure 2: Final Stable situation for a tolerance rate = 0.375, corresponding to 3 neighbours over 8](image3)

![Figure 2: Final Stable situation for a tolerance rate greater than 0.375 and corresponding to 4 neighbours.](image4)

(c) Final Stable situation for a tolerance rate greater than 0.375 and corresponding to 4 neighbours.

Figure 2: 5 populations segregation Schelling model on RePast with density = 0.625 and with 2 values of tolerance rate
the initial population distribution and the final stable distribution are given.

In figure 2, we detail the impact of tolerance rate on the segregation result. In this figure, we extend the original problem based on 2 population classes to 5 classes population. Part (a) describes the initial distribution according to a whole population density equal to 0.625. Part (b) describes the stable population distribution for a similar tolerance rate for each agent, equal to 0.375, corresponding to 3 neighbours on 8. This value is a discrete bifurcation point from where all small additional value leads to a very different distribution. To illustrate this phenomenon, part (c) describes the population distribution for a tolerance rate greater than 0.375, corresponding to 4 neighbours. The final population distribution is completely different than the one in part (b).

Figure 3 describes some singular formation which can appear in very few cases, when we go over the bifurcation point which leads to no global clustering formation, except if some very small cluster kernels appear according to stochastic move spatial conjunction.

CONCLUSION

In this paper, we discuss about the properties of decentralized methods to model the spatial self-organization. These methods are based on the description of the whole system by the interaction systems of individual behaviors. We present some experiments on Schelling’s model where auto-organization emerge from local rules. This model is able to take into account multi-criteria: each population class could be understand as the characteristic of some specific criteria. And the processus leads to simulate the emergent system morphology in order to define its criteria-specific sub-population separation (as an extension of segregation problem toward multi-component morphology development).

REFERENCES

Multiple Virtual Command Centres for Controlling Physically Distant Set of Robots: Next Generation of Telepresence

G. Al-Hudhud
Software Engineering Department.
Al-Ahliya Amman University
email: ghudhood@ammanu.edu.jo

KEYWORDS
Virtual Network, Communication and Control, Telepresence

ABSTRACT

This paper presents the implementation of a 3D virtual environment $VE$ as an interactive collaboration environment between a team of agents. This team can include human operators and robots at different physical locations. The use of a 3D $VE$ as a virtual Blackboard and a virtual world has shown to be useful in order to allow the human operator to: a) observe the emergent behaviour of the robots inside both the real and virtual world, b) allow for user interaction, real time monitoring, and c) qualitatively assess the agents’ behaviour and interactively issue a relevant decision through the virtual blackboard. A set of experiments was being run and aimed at testing the representation of the agents’ behaviours as a combination of the local and global communications in four layers. It is proposed that these four layers can be used to encode more complex behaviours allowing us to test different communication modes and the corresponding behaviour levels. The work presented in this paper shows the integration between virtual networks using blackboard negotiation technique in addition to the flocking algorithm to model the proposed communication modes between human operators and robots. This integration between the virtual and real networks features four main aspects: a) an increased level of presence and immersion within the simulation that helped gain an increased level of understanding of the agents behaviour inside the world; b) enabling a user interaction by flying around the agents and observing and monitoring the agents’ actions, and c) combining the realism and reality of the agents behaviours which is of great importance for ability to control the real physical robots from different physical locations allowed for easy fault detecting and for real-world problems to be tested and seen in an easy and understandable manner.

introduction

Multi-agent systems (MAS) are becoming increasingly important in practical applications. Examples of these applications are data exchange presented in (Nagi 2001), and intelligent autonomous robots presented by Brooks in (Brooks 1990). Despite the disparity of the areas in which these systems appear, interaction is a major feature and can be initialized between agents, whether they are similar agents (i.e. homogeneous MAS) or different agents (i.e. heterogeneous MAS). Lin et al (Lin et al. 2000) describes agents’ interaction as a part of an extended communicative activities; e.g. dialogues, argument or negotiation among agents. Wooldrige states that defining the methods of interaction between agents is like scrutinising the communication protocols and negotiation processes of which agents are capable (Wooldrige 1997). Specifying the negotiation technique in turn relies on the internal structure of the individual components, i.e. the agent’s architecture, as well as on the proposed level of interaction between agents.

Simulation is a very efficient tool for analysing the properties of a theoretical model, therefore, a MAS model provides a basic and practical representation of these theoretical aspects. Due to the complexity of designing a multi agent system (MAS) model, simulation can reduce this complexity by building and testing the system components separately. This step by step building strategy allows the user to test and evaluate individual components during the development stages.

In order to benefit most from simulating a MAS communication model before creating a physical implementation, the simulation must give an insight into the operational requirements. Such requirements include the size of the models and the space they move in. Therefore, visualising the simulation of the MAS model within a virtual environment ($VE$) can actually be more efficient and functional than within conventional simulations.

Simulations within a $VE$ supports simulating the real physical sizes and spaces. In addition, it allows the emerging characteristics to be empirically defined, tested and explained by considering both quantitative and qualitative outputs (Hudhud 2008). The performance of a MAS can be evaluated and assessed within different simulated environments. This implies that agents can be exposed to a variety of different tasks; i.e agents can be created and their performance tested in different environments without an excessive amount of development time. A developed multi-agents system can therefore be used as a testbed for different tasks; e.g. in the deep sea or on a planetary surface.

Recently, research in the area of communication and control emphasising on combining both reality and realism for controlling a set of robots physically existing in a different location (Turner et al. 2006). This implies a set of properties can
be achieved by integrating various tools together, to create a testbed network for the next generation of robot telepresence, which aims to be able to launch a manned spacecraft, twice in a defined period of time, with a skeleton crew of technicians to operate it. Many current UVMs (Unmanned Vehicles) instead of requiring a small operating crew, can require a large team of operators, and be more expensive than the equivalent manned version to run, thus defeating some of the purposes of having UVMs in the first place. Also, in the last 15 years there has been a rapid growth in the study of semi-emergent behaviour when many autonomous agents operate in a closed environment. It is thus imperative if we require emergent behaviour from UVMs we are going to have to change the way telepresence operates.

RoboViz (Turner et al. 2006) is a small step in the way to consider robotic autonomous engines, as connected computational engines, that have to be steered and also have local communication modes towards each other as well as to and from command centres and operators. This way of describing a heterogeneous computing elements being connected with diverse networking standards and wishing to be computationally steered by human operators, has many similarities to traditional e-Science computational steering systems. This paper highlights two main dimensions: a) First, simulation results for multi communication levels of heterogeneous agent-based system; human operators and robots, and b) Second, advanced experimentation study for testing real implementation for multiple command centres and accordingly, describing the individual parts of the testbed being used through the CompuSteer project (Turner et al. 2006).

Background: Common communication Techniques

Common communication protocols use either message sending (FIPA Foundation for Intelligent Physical Agents 2000) or social behaviour-based interaction (Crowther 2004). Meanwhile, negotiation processes by means of message passing is a main interest in the area of MAS. Although different views were presented by Cohen (Cohen and Levesque 1991), Jennings et al (Beer et al. 1998) and Ferber (Ferber 2002), they agreed that negotiation by message passing plays a major role in the theories of communication.

Quintero et al (Quintero et al. 1997) considered co-ordination and co-operation as the main stays of negotiation to be found in most applications of multi-agent systems. Co-operation and co-ordination are used for controlling and distributing tasks between agents as described by Kuwabara et al (Kuwabara et al. 1995). Accordingly, each agent needs to avoid goal conflict while it co-operates with other agents in a way that improves the overall system performance (Beer et al. 1998). Cohen et al in (Cohen et al. 1997) and (Cohen and Levesque 1991) also stated that co-ordination increases the agent’s capabilities to perform the specified task using the grouping technique. The grouping technique implies the agents co-operate by sharing tasks (team performance). Jennings et al (Beer et al. 1998) also identified conflict resolution by co-ordination in a MAS.

Quintero et al (Quintero et al. 1997) describes common negotiation models and maps the solution to a given problem to the co-operative interaction between all agents in the system. Nwana et al (Nwana et al. 1994) described a negotiation model that prevents duplicating the work and coordinates tasks. Jennings (Beer et al. 1998), Wooldridge et al (Wooldridge and Jennings 1995) and (Wooldridge and Jennings 1999) identified negotiation protocols as the set of rules that governs the interaction.

3D Virtual Environments for Simulating Communication

In this respect, the considered simulation tools must: a) give a deep feeling of distances and dimensions on the appearance of objects (geometrical realism), as well as a high level of realism of the agents’ interaction as a group behaviour (behavioural realism). b) define the specifics of the application, simulating a sensor within the artificial visual system, and c) support user-interaction with the agents inside the simulated world (presence and immersion). These features can be obtained by using 3D simulation tools that can be operated on a desktop as well as within a semi-immersive projection theatre.

The realism of this simulation is critical in ensuring that the implemented algorithm during the simulation can be feasibly transferred to a real robotic application. Simulating a MAS network model implies both a realistic representation of physical agents and their environment and above all the realistic simulation of the communication modes and network virtual connections. Therefore, simulations within virtual environments are a major interest within this work because of the additional features that can be obtained that is the building of an environment that mimics, admittedly to a different quality level, the real physical space. This full scale visualisation enables us to test different situations with different scales and dimensions and can enhances the sense of the dimension and sizes of objects inside the environment, see figure 1

The large scale display also offers the opportunity to visualise multiple views at once, see figure 2. This allowed for a quick visual, individual and group, analysis for the quality of the agents’ actions in the virtual world for this large number of agents by allowing us to move closely to the individual agents in different situations.

Regarding the simulating the virtual communication network, significant benefits of using interactive 3D-simulations only occur if the VE assists the user in achieving objectives more efficiently than other simulations would. The simulation tools used in this work allow the human vision system to assess an agent’s behaviour by running the system and projecting the scenes onto a large screen in order to improve the ability to investigate many more design alternatives using different micro structures, that implies testing different agent’s architectures. Therefore, the evaluation process focuses on visually assessing the quality of the agents’ actions and tests different behavioural models to see whether they meet the user’s expectations. Visualising different levels of behaviour
under different conditions allows us to define 'what-if' situations, to discuss times when the system fails, limitations on the number of agents, and sensor range modifications, etc. It provides us with a way to explain the cases that are considered as bottlenecks. This improves the performance by evaluating the emergent behaviour at the system level not just at the individual level over a number of iterations.

The experiments combine the local and global communications into four layers. These layers encode the different levels of behaviour and agent’s architectures within different models: Local communication Model, (LC – Model), Global Communication Model (GC – Model), Local-Global Communication Model (LGC – Model), and Follow Wall Model (FW – Model). The following subsections describes the implementation of each of these models, and by visually testing the agents’ actions, and the outcomes are compared to the user’s expectation from each model.

**Local Communication Model (LC – Model)**

This model implements the flocking rules without any global interaction or knowledge. The LC – Model represents an initial case where agents are expected to only communicate with nearby agents to coordinate movements. Therefore the only inputs for the agents in this model is the received information via the agent’s sensor and they are not assigned any tasks. The expected behaviour from this model is to obtain the natural behaviour of a flock. The agents are expected to show this reactive behaviour that represents animal intelligence in avoiding each other and moving in a group.

**Figure 2:** Leicester Reality Centre, De Montfort University, displaying multiple views simultaneously.

**Figure 3:** A set of agents are moving within a team and influenced by the cohesion force and the alignment force.

Running this model results in a lower level of behaviour and allows for the reactive components of the agents to be active. This model allows us to test the system’s micro structure; i.e. the reactive component of the agent’s architecture.

**Figure 4:** A team of agents moving forward, the team members split into two teams as they encountered the wall.

The tests show a high level of co-ordination as agents were able to move in groups with signs of natural behaviours. For
example, the flock can split up into two smaller flocks to
go around both sides of an obstacle and then rejoin once
past the obstacle. In addition, a dynamic leadership is also
achieved as any front agent can lead. So, if the leader should
get trapped between other robots, any front agent will take
over the leadership and the nearest agent will follow.
Figure 3 shows a set of agents represented as 3D-robots. They
move randomly following the first three rules of the algo-
rithm. An advantage of implementing the flocking algorithm
is that agents move through different routes. This is quite
useful for some applications such as passive searching or res-
cuing tasks.

**Global Communication Model (GC – Model)**

For this model the ground based controller (human) broad-
casts messages, via the keyboard, to all registered agents
within the system. These messages include explicit instruc-
tions to allocate tasks for a set of agents. When these agents
read the message, they are grouped into a team; the concept
team forming. In terms of the flocking rules, each agent
only considers the collision avoidance rule. In this context,
an agent’s internal system sets the weights that control the
strength of both alignment and cohesion force to zero. The
effect of applying global communication in this way is equiv-
alent to the effect of applying a global alignment force, i.e.
all agents will align their velocity in the direction of the target
position. According to this model, an agent is expected to ac-
tivate only the cognitive component of its architecture. This
component allows an agent to possess a cognitive knowledge
structure and in this case each agent of the group is supposed
to possess a joint mental state that leads to a joint intention;
i.e. performing the common goal.

This model is tested by running the system with a specified
number of agents, (five agents for the first trial), and allow-
ning the GBC to interactively issue a message to the agents
using the keyboard as an input device. The agents inside the
virtual world receive the message and interpret the contents
of the message. The agents first issued messages to the GBC
to report their status at each time interval $\tau$. For example, an
agent is happy to perform the task if $(T <= C)$, also it is
a little bit “Tired” if $(T - C <= 20)$ 
$^2$. If $(T - C \approx 1200)$,
this means the elapsed time of about (2 minutes) since dif-
ficulties started to arise, but still an agent will try to per-
form the task. But if $(T - C >= 2000)$ then it is likely to
be trapped somewhere, and status ‘lost’ is reported to the
GBC. If the $(T - C >= 6000)$ implies that the elapsed time
say (5 minutes), an agent will report a virtual death status
‘dead’ to the GBC. For both ‘Lost’ and ‘Dead’ status, an
agent will do the following actions: It reports its status to
the GBC via the external-blackboard and discards the spec-
fied task; sets the weight of this rule to zero. It then moves
randomly until it receives an alternative command from the
GBC, who can monitored the situation. This implies that an

$^1C$ is the expected time for task completion, while $T$ represents the
elapsed time since the agent started the task.

$^2T - C$ is the time elapsed since the agent started to encounter problems
in performing a specified task.

agent is unaware of other recipients of the same message.
The test has shown that an advantage of this is that the action
may not be discarded from all the team members.

**Local-Global Communication Model (LGC – Model)**

According to this model, agents are expected to be able to
communicate with each other as well as with the ground
based controller GBC; i.e. all the communication levels are
implemented. All agents are supposed to be grouped in teams
as a consequence of issuing a group command by the GBC.
So this model exploits agent’s reactive and cognitive compo-
nents represented by the HBDI integrated with the BB in one
architecture.

The test considered the scenario where a set of agents are as-
signed a task, e.g. they are given a target location, in the up-
per left in figure 2. By assigning the virtual task, each of the
agents in the group is expected to: a) compute the expected
distance to the task, b) estimate the time to finish the task,
and, c) issue messages to the GBC reporting these informa-
tion as a status ‘HAPPY’. Each agent is expected to report its
status depending upon its dynamically changing beliefs. The
reported status is a result of activating the deliberative com-
ponent of the agent’s architecture together with the reactive
component. In other words, an agent uses all the informa-
tion available, not only those related to the local reactions
but also those related to the GBC’s commands. On arrival,
agents are to circle around the target and continuously mod-
ify their weight for the blackboard negotiation rule, in order
to avoid getting too close to the target.

![Figure 5: Team members are circling around a target posi-
tion.](image)

Running this model has shown interesting results across the
micro and macro levels. Regarding the macro structure, the
agent’s actions have met the expectations where the deliber-
ate component in the agent’s architecture works syn-
chronously with the reactive component. This test allows
us to compare the different behaviours based on different
knowledge structures. The emergent behaviour depends on
the cognition intelligence as a prerequisite for the deliber-
ation, see GC – Model. Regarding the macro level struc-
ture, the visual tests assists in building the new model that
reflects the features of both models into one, namely the
The visual tests played a major role in evaluating the group communication as it allows for interactively monitoring the agents’ actions in real time. The four layers of communication (three layers for the flocking algorithm plus one layer for the blackboard rule) are all implemented.

In addition to the visual outputs, the numerical outputs can also be analysed, for example, the interaction weights and positions. This information together with the visual experiments allowed us to identify the effect of the cohesion force as a binding force that affects the agents’ progress, where the user is able to numerically analyse the weights. This has been significantly useful when the visual test showed slow responses in the agent’s progress when combining the flocking rules with the blackboard. The results of running this model show that the user is able to get comprehensive information from agents in the form of actions and written messages. The user can determine the conditions that increase the chance of completing the task despite difficulties that may arise, for example, when any agent fails to perform the specified task and other team members will then continue to perform the task. Practically, this has been implemented in the same manner as described in section.

Oscillate State Detection and Follow Wall Mode: FW – Model

The FW – Model resolves the conflict that can arise when an agent needs to reach a specified point that lies behind a barrier or a wall. The ‘Wall’ problem implies that an agent would turn to avoid the wall then turn back heading toward the target so leading to oscillation. This can also happen, for example, when an agent’s sensor reads three consecutive similar values for the identity of the detected object within the perception zone of the collision avoidance rule, (Hudhud 2008).

The basic idea behind the FW – Model is to let a meta-level component run the world model faster than real time to make predictions of future states. When the meta-level component detects that the system has specified an undesired action, it modifies the decision made to produce a new actual action in order to avoid reaching this state. This model enables the agent to detect the oscillation state and then allows the agent to switch to another mode. The first is the standard LGC – Model, whilst the second is the Follow Wall Mode or FW – Mode. The FW – Mode is used as a recovery mode and allows the agent to move smoothly alongside the wall until it can turn again toward the target. Once an agent has passed the oscillation state, it switches back to the standard LGC – Model.

The FW – Mode is implemented as follows: Firstly, each agent performs a pre-collision detection test using its current heading. Secondly, it computes a new heading according to the communication algorithm explained in section ???. The agent then will use the new heading to perform a post-collision detection test before changing its current heading. An agent will not change its heading unless there is an object straight ahead. For the cases when an agent detects three different objects this model may still fail as an agent may still show oscillation whilst trying to avoid these objects.

![Figure 6: An agent moving toward a target. The target position is behind the wall so it follows the wall until it is possible to turn back to the target position.](image)

A Collaborative User and Robot Environment Network Testbed

This integration between the virtual and real networks features key practical aspects. First, an increased level of presence and immersion within the simulation helps gaining an increased level of understanding of the agents behaviour inside the environment. Second, the virtual network enables a real-time user-robots interaction by observing and monitoring the robots’ actions and analysing the information sent by the robots to a human operator. Third, The proposed implementation of the virtual network combines the realism and reality of the agents-based communication modes and the corresponding behaviours. This can be of great importance for ability to control the real physical robots from different physical locations allowed for easy fault detecting and for real-world problems to be tested and seen in an easy and understandable manner. The following sections presents a study for this kind of implementation including the response time for multiple connections, building a testbed for these multiple connections, and the choices for the testbed.

Loss, Latency and Multiple Connections

The networking involved within a telepresence robotic interface has two key components; the control network predominately sending instructions from an operator to the robot, and the information network transmitting data, often predominately video (especially when considering required bandwidth), from the robot back to an operator. Given reliable

---

3 An operator is defined here as a close connected hands-on controller (note there could be more than one), and a user is defined as any human with the ability to take control directly or indirectly.
communication channels, and dedicated links with known, if not actually removed, latency then the command-and-control system, within this telepresence, is tightly controlled and becomes an easy to understand loop.

We can now consider the time delays involved in this ideal system. Video and diagnostics information from the robot are transmitted over a time delay, $\delta v$, and after a response time from the operator, $\delta u$, an instruction is sent to the robot over a further network time delay, $\delta c$, and acted by the robot after interpretation with a final delay of, $\delta r$. This cycle repeats with trained operators controlling the robot with an extended reaction time of,

$$rt = \delta v + \delta u + \delta c + \delta r$$

A trained operator can use this kind of simple system with a high degree of efficiency, even experiencing extra input for example robotic haptic feedback (transmission of force/touch direct to vibration/force feedback devices) over quiet a range of latency times. As latency times increase efficiency decreases, but is in some respects a controlled degradation of service. An issue arises when extra influences occur that can make the control or robotic steering operation difficult or even impossible, which we will now list and consider.

The first influence we will describe is variable latency. It is well known that given a fixed latency, even if it is very large for example from Moon to Earth (up to 2 seconds time delay with direct visibility but raising to 10 seconds if a data relay satellite is used), over a training period, preferably offline, an experienced operator can reliably control an on-line robot at a constant level of efficiency. This is unlikely to approach the same efficiency of an operator with direct control, but what is important is that there is a constant percentage efficiency value. Networks are not usually built like this (unless for example they are dedicated switching systems [dark light fiber]) and their latency will fluctuate over time. Now the extended reaction time is a function of time $t$,

$$rt(t) = \delta v(t) + \delta u + \delta c(fn(t)) + \delta r$$

These fluctuations in latency make the task an extra degree harder for the operator, potentially drastically reducing the amount of useful work achievable. A simple approximation is to model these fluctuations around Gaussians or Poisson distributions, with known mean values, but this may not take into account burst temporal delays, as networks have non-homogenous effects.

A second influence of a standard network is the probability of loss and corruption (e.g. data convolution and additive noise) within the transmitted data that is a natural occurrence. Again, for an operator this introduces uncertainty and thus reduces performance. So for each of the networks we have probabilities $p$ of corruption or loss for a particular element of size $s$, which again is time dependent,

$$p_s(v(t)), \quad p_s(c(t))$$

Although latency and loss can be modelled by a generic corruption/delay possibly in threshold filtering the convolution of the transmitted stream $t$ with a point spread function $p'$ and adding noise $n$. As networks are non-linear in (packet) data recovery this is only a crude approximation.

$$s = t \otimes p' + n$$

If $s$ is very different from $t$, i.e. above some threshold $T$ then the resulting data can be unrecoverable.

$$\text{Fail} : \{ T > |s - t| \}$$

A third problem introduced is the fact that a single operator controlling only one robot is inefficient, and the introduction of multiple robot control is essential. This aids both efficiency, as well as allowing new joint collaborative actions to occur between the robots, that may not have been possible otherwise.

Finally we would like to add specialist expertise and guidance, for the operator, involving other users, each monitoring the progress as well as being able to act, take over control, or simply suggest alternative courses of actions. This brings together a requirement for a set of multiple command and control centres with multiple robots over standard networks. This brings a need for a third network to be introduced that connects command centres together creating a shared collaborative space.

**Building a Testbed Communication System**

To consider these issues we propose to consider a real robot network with a set of real telepresence operators distributed using different packet style systems, each considering the requirements for multi-way communication, loss and latency.

**Choices**

**UDP/IP**

The network connections will consider UDP/IP to transmit continual streams with no concern for full error corrections and recovery, by inherently not allowing packet resending. In a real-time system, hearing or seeing data which should have been transmitted some time previously, if missed is possibly too late to be useful. As temporal latency increase for a packet, a practical solution is to introduce larger than average packet buffers at the receiving end - which will also increase the mean latency value. There is a continual battle of trade-offs between mean latency and amount of packet loss. Increasing the mean latency, by having a large packet buffer, reduces the number of packets lost, but also increase the reaction time $rt$ for the system.

There is a difference of opinion as to whether the data stream, consisting of commands, should be error correction for missing packets, e.g. TCP/IP, or if even simple command streams can also have lossy influences that are acceptable.

---

4In fact many operators may be required to operate one robot
Multicast transmission
Any system should be scalable, and the usual unicast (one-to-one) communication systems have an $O(n^2)$ scaling issue as the number of robots and/or the number of command centres increases. [ref wikipedia unicast and broadcast]. Using multicast the aim is for a single packet to be optimally transmitted to a range of addresses, meaning along any specific connection no more than one copy will need to be transmitted. This is essential to cope with a large increase in number of robots, operators and command centres.

Compression Standards
All data streams that do not need to be transmitted faithfully, for example video and audio can be also lossy compressed. For a 25Mb/s connection (low average guaranteed within academia), with lets say 20-30 video streams each should aim to be well under 1Mb/s in order to accommodate the outgoing network as well as include audio and other data streams. Compression can consider all forms of quantisation: temporal (number of frames per second), spatial (x-y resolution of the image) and within the compression codec for example quantising in frequency (for example in Discrete Cosine Transforms, or Wavelet transform codecs).

Intra-Robot communication based on SpeechActs
The language required for robot to robot requests needs to be formal in structure. This allows for analysis prediction, and behavioural modelling. This also allows localised communication, for when two robots physically (or visually) see each other and can pass intentions, behaviours and states of being.

Semi-Autonomous robot behaviour
If we can not guarantee data being transmitted arriving, nor guarantee the exact response times to request, commands can not be at the microscopic detailed level. Thus, robots need also semi-autonomous behaviour for two main reasons; one so that they can continue to operate safely (negotiate doors, walls etc.) when there is no direct assistance, and secondly so that together, by working in teams, the emergent behaviour can allow functionality that is too complex for a single operator to manipulate. The system currently used is based on flocking rules and known blackboard intents.

No single point of failure
An extra aim should be for the whole system to operate with any number of destroyed links. This means there is no single point of failure that would jeopardise a large part, or all, of the network. Although certain nodes in the system will be affected, the complete system should degrade gracefully under failure.

Small transmission footprint (Linux)
To accommodate some of these specifications the codecs and standards for the Access Grid (www.accessgrid.org) will be considered for the video and audio stream transmission. For transmission to reduce the footprint on a robot (assuming linux based infrastructure), the Java Media Framework library (JMF), can video capture (via for example from USB), encode and transmit compatible RTP (Real-Time Protocol) which can use the current Access Grid standards (vic and rat). For clusters of robots it would be sensible to connect the streams through a bridge, although this introduces a local single point of failure. Multicast can then proceed from that point if available (in test systems firewalls and networking protocols can cause other unknown single points of failure.)

Steering Architecture
With this system every command centre will receive all the video streams (possibly in stereoscopic projection if required and available) from all the robots, as well as from other command centres. This gives a specified experience of latency and thus efficiency to control the robots. Each robot can be considered as a computational engine, considering its next actions based on both higher level tasks from the command centres, as well as, low level information from its environment.

Multiple users can try and steer the same robot, which in temporal latency issues can occur no matter how well practiced operators are. The robot needs to take a decision based on the multiple instructions, possibly contradictory, as well as of course the local environment. For example this means that two instructions, one to turn right and the other to turn left, to avoid a hole will not result in the robot carrying out the average of the two instructions. Diagnostic status information returning from the robot, indicate the current course of action, back to the operators and thus with communication across command-centres difficulties can be alerted.

The users still need protocols to operate but as the command centres are video and audio linked inter-command centre steering can be done (initially by shouting). This is not ideal but for the first generation of test-bed this has advantages of procedural optimisation as well as adding layers of computational blocking at a later stage.

Facilities at Command Centres
All the standard benefits from an Access Grid (Collaborative Working Environment) structure are expected within any command centre (which may not be the case in the future). This helps to increase the level of presence experienced by the users, so other command centres appear in similar co-located spaces as well as co-located with the spaces inhabited by the robots. This includes;

- Large life-size projected screens (for full size viewing)
- Directionless remote microphones - so there are no cables or even radio-tags required
- Echo cancellation system – so headphones are not needed
- Shared data sets as well as the media streams – so environments can be considered integrated
A final issue to be considered is that of stereoscopic projection, that is available on a subset of conference rooms. This allows for 3D-stereoscopic robot vision to be transmitted to any or all of the users and operators, as well as data sets - for example building schematics - to be shared in 3D at different sites.

**Conclusion**

This paper presents the 3D VEs as an interactive simulation in order to observe the emergent behaviours when implementing a MAS communication model. It introduces the use of the 3D interactive simulation to enable designers to operationalize the theoretical concepts for empirical studies. The 3D simulation tools used in this work offer sufficient techniques that allow for perspective views of the agents and the environment in real time. It also allows for user interaction, real time monitoring, and qualitatively assessment of the agents’ behaviour. Therefore, the focus was on assessing and testing the representation of the agents’ behaviours as a combination of the local and global communications in four layers which encode different levels of behaviour. It is proposed that these four layers can be used to encode more complex behaviours allowing us to test different communication modes and the corresponding behavioural levels.

The paper also presented that the integration between the virtual and real networks. The study presented by this paper has been shown to be useful for increasing the level of presence and immersion within the simulation which helps gaining an increased level of understanding of the agents’ behaviour inside their environment. The proposed virtual network enables a real-time user-robot interaction and analysing the information sent by the robots to a human operator. The proposed implementation of the multiple network connections combines the realism and reality of the agents-based communication modes. This can be of great importance for ability to control the real physical agents from different physical locations.

**REFERENCES**


AGENT BASED SIMULATION
AGENT BEHAVIOUR
Modeling the Emotional State of an Agent Through Fuzzy Logic With Reference to the Geneva Emotion Wheel

William F. Blewitt
Aladdin Ayesh

Centre for Computational Intelligence
De Montfort University
The Gateway
Leicester, UK, LE1 9BH
E-mail: {wblewitt|ayesh}@dmu.ac.uk

KEYWORDS
Emotion modeling, Fuzzy Logic, Affective Agents.

ABSTRACT
The issue of accurately modeling human emotions in agents has been the subject of extensive debate for some decades. This debate, complicated as it is by constant psychological arguments about the nature of emotions and their affects on behaviour, has given rise to many different schools of thought within the greater, overarching subject of affective computing. Our work centres around a paraconsistent logic system rooted in fuzzy principles, its goal being to more accurately model the emotional state, and changes of the emotional state, through usage of an emotional circumplex. Within this paper we intend to outline our system for modelling emotional state and changes to emotional state. We shall include an outline of the psychological basis for our work, and discussion of primary prototyping of the proposed system.

INTRODUCTION
The accurate modelling of the emotional state of an agent, and changes to emotional state, has long been the subject of extensive research (Ahn and Picard 2006, Ekman 1992, Naqvi et al 2006). This work has varied in scope from the philosophical questions about why to make an emotional agent, to the sociological questions about what impact emotional agents have on human-computer interactions. In computing sciences, such debate has tended to revolve around the quintessentially practical questions regarding the mimicry of emotional responses and modelling of emotional state.

Within psychology, there are many schools of thought regarding the nature of emotions and how they are best modeled. We have performed an extensive review of these approaches, and further to this concerned ourselves with how they have been previously implemented in the field of artificial intelligence.

Recent publications in psychology have expressed a growing view that, rather than emotion being an impedance to human creativity and decision-making, it is indeed an asset, if not a fundamental element of the cognitive process (Muramatsu and Hanoch 2005, Perlovsky 2006). As this viewpoint begins to garner wider acceptance, it is expected that a renewed interest in emotion modelling and representation, from an academic standpoint, will follow, along with an exploration of emotion models that are not inherently linked to elementary control.

Our model is built upon the viewpoint that the emotional state model itself should have the capacity to be wholly independent of the elementary control systems of the agent, instead being a semi-isolated construct to determine ongoing emotional state.

Within this paper we shall present some preface, discussing currently accepted approaches to modelling the emotional state of an agent, and how this work has fed into our own. We shall then proceed to outline the psychological emotion model we have selected. We shall present a method of applying fuzzy logic to the model itself, and some illustrated examples of how this interface might function.

PSYCHOLOGICAL BACKGROUND

Within the various fields of psychological research, two schools of thought appear to dominate the debate regarding the nature of emotions, and how they are best modelled. From a philosophical perspective, the nature of their divergence and their theoretical differences are of great importance; from a computing sciences perspective, however, their differences lie entirely in the nature of the models they propose.

The view of emotions as an evolutionary construct was initially proposed by Darwin in 1872 (Darwin 1872). Over the past century, this has ultimately given rise to a school of thought which maintains that there are several fundamentally defined emotions, and that any given emotional state is a function of, or defined by, these emotions.

The exact number of ‘fundamental’ emotions widely varies. Plutchik first proposed his system of emotion classification in 1980 (Plutchik 1980), containing eight fundamental emotions. In contrast, Ekman proposed a system consisting of six fundamental, or basic, emotions in 1982 (Ekman 1982). Within this school of thought, the maximum number of basic emotions is generally thought to be fourteen (Scherer 2005).

Following on from the definition if basic emotions comes the definition of more complex emotions. Often these categories are divided using nomenclature indicating primary and secondary emotions as in the structure proposed by Parrott (Parrott 2001). In general terms, it is the view of this school of thought that the sum of human emotional
experience can be defined as a function, or construct, of less than a dozen named emotions (Ekman 1992, Scherer 2005). An alternative to this view proposed by Wundt in 1904 suggested that emotions could be better defined in the context of experience than crisp linguistics (Wundt 1904). Research based on this principle has, as with the Darwinian view, given rise to many varied schools of thought following the same fundamental idea.

In Wundt’s original model, emotional state was represented in terms of three facets of experience which he labelled ‘pleasantness’, ‘approach’ and ‘arousal’. He asserted that any individual emotion would be better modelled in the context of relative magnitudes of these facets of the emotional experience than through verbal labels.

Subsequent to Wundt’s original work, significant research has been performed regarding this idea of a ‘dimensional’ emotion model. In many cases it is common for the third axis to be ignored and, instead, for proponents of this view to model emotions in the context of ‘valence’, which might be seen as a clearer definition of ‘pleasantness’, and arousal. More recently, however, it has been suggested that these views are not necessarily mutually exclusive. Russell produced a circular model of emotions outlining the position of what he believed to be fundamental emotions in terms of relative values of what were effectively arousal and valence (Russell 1980). Indeed, a geometrical approach to the views espoused by Ekman and Plutchik may be viewed as a very sensible way to continue (Scherer 2005). It is upon this idea of a hybrid of the two major psychological schools of thought that we have focused our attentions. In particular, on the works of Scherer in the context of his geometric model of emotions in the context of ‘valence’ and ‘control’ (Scherer 2005).

ALTERNATIVE DIMENSIONAL STRUCTURES OF THE SEMANTIC SPACE FOR EMOTIONS

The Geneva Emotion Wheel (Scherer 2005) proposes that the emotional state of a person may be determined through a geometric structure comprising of sixteen base emotions at differing levels of magnitude. Scherer’s investigation of this concept was based around the semantic identification of the position of individual emotions along a circumplex considering magnitudes of conductivity and control. In his paper, he outlined the idea that the variable magnitudes of these key emotions could be considered an acknowledgement of other emotions not considered fundamental. For our purposes, this provides an emotional statespace comprising of sixty-four discrete emotions from which to derive the emotional state of an agent governed by this system.

A graphical representation of the Geneva Emotion Wheel is included as Figure 1 for illustrative purposes, adapted from its original presentation by Scherer (Scherer 2005). It is conceptually difficult, when approaching this problem from a traditional logic standpoint, to account for the emotional ‘dead space’ interwoven between discrete emotions. That said, the graphical representation as presented by Scherer was conceptual, rather than geometric, and our interpretation of the model accounts for the issues mentioned.

A major consideration in our selection of this model was the fact that the emotions themselves are removed from the experiential factors that define the model. In the case of the Geneva Emotion Wheel, these factors are Control and Valence.

The existence of these factors enables us to remove the emotions from any direct connection with the systems actions or perceptions, an important concept given the intent of our work. Furthermore, Scherer comments in his work that when overlaying the semantic statespace models of his own statistical analyses with those of Russell, there could be defined a direct relationship between both his Valence/Control topography, and Russell’s Conductivity/Arousal topography.

![Figure 1 Geneva Emotion Wheel (Adapted from Scherer 2005)](image)

**MODELLING THE GENEVA EMOTION WHEEL THROUGH FUZZY LOGIC**

Modelling this chart in a fuzzy form is a conceptually simple exercise. We highlight the ‘control’ and ‘conductivity’ axes, and consider the wheel as any other two dimensional plane. Each grade of each emotion is considered to have a membership value of 1 at the coordinates the wheel states they occupy, relativistically speaking.

Further to this, each emotion shall be modelled by two membership functions, both with maxima located across the same circular region. One membership function shall exist parallel to the ‘conductivity’ axis, and the other parallel to the ‘control’ axis. These shall define the system inputs.

The system outputs shall be the sixteen named emotions on the Geneva Emotion Wheel. Each named emotion will have four membership functions associated with it, once for each grade in the model.

Our system then receives a minimized rules base linking memberships of Control and Valence with memberships of the named emotions. Applied uncertainty to the membership functions within the Control and Valence variables provides noticeable overlap between similarly located emotion grades, meaning that for any point on the circumplex there shall be a dominant and several subdominant emotions making up the emotional state.

Through this we will be able to model the system experiencing varying grades of multiple emotions.
simultaneously, while preventing the issue of emotional dead space directing the system to believe that it has no emotional state.

Figures providing a graphical representation of the input and output membership function archetypes are provided in the next section where we outline a practical implementation of the fuzzy inferencing system.

In terms of changing the emotional state of our system, we first recollect that the circumplex accounts for four axes, not two, in its derivation (Scherer 2005). Our abandonment of the axes ‘arousal’ and ‘conductivity’ in terms of defining the fuzzy membership grades of each emotion does not require us to abandon them as concepts in the context of a change in emotional state.

Let us consider an event which prompts a change to the emotional state of our system. The specifics would naturally be entirely contextual and based on the purpose of the agent, but in general terms we can consider that any event with an emotional component behaves in the same way.

Such an event would impact the qualitative levels of control and valence of the emotional state of the agent. This would by necessity change the emotional state of the agent to some alternative value. The magnitudes of these changes should, in order to maintain consistency, be related to the current magnitudes of both control and valence – how we react to a given stimulus, both emotionally and physically, must naturally have as a component our current situation. The mathematical discussion regarding this concept is included below.

Within his article, Scherer noted a relationship between four facets of emotional experience – not only valence and control, but arousal and conductivity (Scherer 2005). Investigations are ongoing into the idea of a four-factor formula being applied to individual experiences in order to determine their impact on emotional state, as some events might naturally be more easily identified in terms of arousal or conductivity than valence and control. For the purposes of this paper, we focus entirely on the factors of valence and control.

Some issues identified early in the development of this model were the possibility that areas of emotional dead space might cause issue, but a solution to this problem was proposed. Additionally, this model provides little scope for the possible representation of ‘mood’.

**Mathematical Formulation of the Model**

In discussing its mathematical formulation, the circumplex is a two-dimensional universe within which is defined the emotional state of the system as a point with two coordinates, $x$ and $y$. This emotional state at a given time $t$ shall be defined as $E_t$.

$$E_t = (x, y)$$

The selected emotions within the circumplex are each represented by two fuzzy membership functions: one for control, and one for valence. For a given emotion $e_t$, its membership grade is determined by the Mamdani FIS based upon the coordinates of the point in the statespace defined by Control and Valence.

The Fuzzy Inferencing System outputs the non-trivial results for each grade of each named emotion. These non-trivial membership grades will be used for all of the agent’s emotion-based decision-making, and determine to a greater or lesser extent the behaviour of the system. In a situation where the emotional state $E$ is changed by exterior environmental conditions, the behaviour of the system will likewise change.

Such occurrences as effect the emotional state $E$ will have predefined functions relating the magnitude of their effect to the current values of $x$ and $y$, either through crisp links or functions, or fuzzy links or functions. Let us consider an event $J$. Like $E$, $J$ will have two components, each of which shall be a function, as follows:

$$J = (f_x (x,y) : f_y (x,y))$$

$$f_x (x,y) = dx$$

$$f_y (x,y) = dy$$

These are represented as functions related to $x$ and $y$ in order to allow us to demonstrate subtle changes in emotional state, and general emotional tendency based on current emotional intensity. Essentially, the higher the level of emotional intensity, the less likely a single experience will be to significantly lower that intensity.

Following on from this, a new emotional state, $E_{t+1}$ can therefore be calculated in the following manner.

$$E_t = (x_t, y_t)$$

$$E_{t+1} = (x_{t+1}, y_{t+1})$$

$$x_{t+1} = x_t + dx$$

$$y_{t+1} = y_t + dy$$

After these calculations have been performed, the respective membership grades of the specific emotions are recalculated based on these new values for $x$ and $y$, and the behaviour of the system is modified accordingly.

In the subsequent practical implementation, it was assumed that any changes to the relative values for Control and Valence were crisp. Future work will include an in-depth analysis into the effects upon the model when those changes are not crisp, but rather fuzzy.

**PRACTICAL IMPLEMENTATION**

The model was implemented using the MatLab Fuzzy Toolkit. Two inputs were declared, values from 0 to 200, to represent the Control and Valence factors of the system. Thirty-three membership functions were declared for each input variable, each of which was related to two grades of named emotions, save for the thirty-third which was present to define Apathy. A graphical representation of the Control Variable’s membership functions is included as Figure 2.
As the model is symmetrical, the membership functions connected to Control and Valence are identical. Essentially, overlaying the membership functions defining Control and Valence generates a topography of membership grades for every tuple represented by the circumplex. Seventeen output variables were declared in the system. One for each of the named emotions, plus one for apathy, which we must assume is the null emotion at the centre of the model. All variables save Apathy were given four membership functions, one for each grade present in the Geneva Emotion Wheel. Figure 3 outlines the membership functions of the output variable Hostility.

The Mamdani FIS was provided one rule to define each grade of each named emotion, plus one rule to define apathy – sixty-five rules in total. We consider this a rules-light system, given the nature of the model we are representing – that said, we are currently investigating ways in which to optimize the rules list and potentially reduce the number of required rules. We now include an overview of the FIS functionality from a user perspective. Figure 4 demonstrates the front end of the user interface through Matlab’s fuzzy toolkit.

The user can select any values for Control and Valence between 0 and 200 and input them through the rules interface. The FIS then determines the membership values for each emotion within the system; we concern ourselves, however, with the named emotions that have non-zero values.

Figure 5 outlines an example taken from a screenshot of the system functioning, but amended to exclude all emotion facets with a zero result.

This example outlines an agent whose emotional state can be summarized as comprising of Disgust and Guilt, but a mid-level intensity of both.

As a second example, demonstrating the change in emotional state possible through simple changes to the values of Valence and Control, Figure 6 shows the non-trivial results of higher valence and lower control.

Note that in this case we again have two named emotions forming the general emotional state, Interest and Hope, with upper-middle intensities of both. We should also note that both Figures 5 and 6 came from screenshots of the system in action, but were edited to remove large numbers of zero
results and to make them more practically serviceable as figures. For the sake of completeness, Figure 7 shows the screenshot that Figure 5 was taken from. This would be the screen a user would see and interface with when interacting with the system.

Figure 7 Unedited screenshot of the system output from which Figure 5 was derived

CONCLUSION

Within the paper we have provided a brief background to the research we have undertaken, to place it in context. We have discussed the psychological principles which impact computerized representations of emotional state. We have proceeded to outline a fuzzy logic representation of the Geneva Emotion Wheel, and discussed our reasons for pursuing that model in particular. We have included mathematical representations of the model for purposes of clarity. We have then shown the implementation of this model through Matlab and provided examples of its use and user interface.

Our future work shall include rigorous testing and comparisons between this model and alternative fuzzy logic-based emotion modeling methods. We shall investigate various methods of interfacing the model with testing agents. We shall also attempt to more accurately calibrate the fuzzy membership functions defining each individual named emotion through statistical and empirical means.

In addition to this, we will investigate the impact of fuzzy changes to emotional state, and the uncertainty this would introduce to the system. It is the goal of this research to provide one more piece in the growing puzzle that is the representation of human emotions in artificial agents.

REFERENCES


Scherer, K. R. 2005, "What are emotions? And how can they be measured?", Social Science Information, No. 44, volume 4, pages 695-729, 2005

LEARNING AGENT-BEHAVIOR FOR AGENT-BASED SIMULATION USING GENETIC ALGORITHMS

Benjamin Wolters, Timo Steffens
Fraunhofer IAIS
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
email: timo.steffens@iais.fraunhofer.de

September 9, 2008

KEYWORDS
Agent-based simulation, high-level petri nets, learning, genetic algorithm

ABSTRACT

Agent-based simulation is a convenient method to analyze different strategies for specific scenarios, such as evacuating a crisis region or driving a convoy through unsecured areas. The quality of the simulation results depends heavily on the accuracy and diversity of the modelled agent-behavior. This paper presents a method to learn such agent-behavior automatically in order to acquire near-optimal strategies.

INTRODUCTION

Simulating scenarios such as evacuations or convoy transports are an important factor to ensure the success of such operations (Yildirim et al. 2007). When agent-based simulation is used, each active entity is modelled as an agent with its own behavior (Hgelmeyer et al. 2006). The strategy of one team of agents (often called blue team in operational terminology), such as the medical units, is pit against the strategy of another (red) team, such as attackers. The goal is to find a strategy for the blue team that reaches its goal despite the attempts of the red team to hinder the blue team. Thus, the practical value of the simulation results depends heavily on the following questions: Does the blue team act as good as possible? Is the simulation run with a wide variety of strategies in order to find the best one? Only if both questions can be answered positively, it is a valid claim that the strategy of the blue team has a high success probability.

However, models built by humans have their limitations. It is not guaranteed that the modelled strategy for the red team is nearly optimal. And it is also not guaranteed that the modelling human thinks of all possibilities. This paper proposes to augment the strategies built by humans with strategies that are learnt automatically using techniques from machine learning. More precisely, we present a genetic algorithm that can built strategies (i.e. agent behaviors) from scratch or optimize existing ones. Genetic algorithms have the inherent property to explore large hypothesis spaces. Thus, it is a promising approach to find agent behaviors that human modellers would not have thought of.

The remainder of the paper is organized as follows. In the next section, we outline the representation language for agent-behaviors which is an important basis for the learning method. Section 3 describes the genetic algorithm that learns new agent behaviors. Experimental results are presented in section 4. Finally, the last section concludes and outlines future work.

PETRI-NET-BASED AGENT-BEHAVIORS

Several specification methods for agent behaviour apply a process-oriented view, for example state charts (Obst 2002), Petri Nets (Steffens and Hugelmeyer 2008), or Markov Decision processes (Kaelbling et al. 1996). Like other simulation specification languages such as Maisie (Bagrodia and Liao 1994), SILK (Healy and Kilgore 1997), SimScript III (RICE et al. 2005) or SSJ (L’Ecuyer and Buist 2005), Petri Nets and their extensions inherently support parallel processes. Additionally, Petri Nets allow to specify different abstraction levels (Vojnar 1997) and can be used to specify agent behaviour as well as implement it. This paper proposes that Petri Nets are a promising representation language in order to learn agent behaviours. The presented approach applies a genetic algorithm on graph structures that are represented by an extension of Petri Nets, called Language for Agent-based Modelling of Processes and Scenarios (LAMPS) (Steffens and Hugelmeyer 2008).

Similar to Petri Nets, LAMPS defines places, transitions and relations that constitute a bipartite graph (Vojnar 1997, Jensen 1992). Places hold typed, structured tokens. Informally, a token in LAMPS can be viewed like an object with attributes. As opposed to finite state machines, places can contain several tokens, and several places can contain tokens at the same time. Thus, the overall state of a LAMPS graph is denoted by the set of all actual tokens. For performance reasons, not only tokens, but also places are typed in LAMPS.
Transitions consume tokens from places and generate tokens for other places. In LAMPS, transitions correspond to agent actions. Each transition contains a set of rules that define in which situations the transition is executed. These rules process tokens and their structure. Transitions and their rules are processed in parallel. LAMPS relies on a global clock that defines rounds (Richter 1999). A transition may only be executed once per round.

Relations are directed and link places to transitions. Figure 1 shows an example of a LAMPS graph.

Formally, a LAMPS graph is an ordered pair \((P \cup T, R)\), with places \(P\), transitions \(T\), and directed relations \(R \subseteq ((P \times T) \cup (T \times P))\). When used for specifying agent behaviours, it is often useful to define specific input places \(I \subseteq P\) and output places \(O \subseteq P\). At the beginning of each round, tokens are fed into the input places by the simulation environment, and at the end of each round tokens are read from the output places.

The signature of a transition consists of the places it reads from and the places it writes into. That is, the signature of a transition \(t\) is formalized as \(\text{sig}(t) = (I, O)\), with \(\forall i \in I: (i, t) \in R\) and \(\forall o \in O: (t, o) \in R\).

Now, an important aspect of Hierarchical Petri Nets such as LAMPS is their ability to scale the granularity or abstraction. Conceptually, a LAMPS transition can recursively consist of a LAMPS graph. That is, the signature of the transition is used as the signature of a LAMPS graph. Tokens that are consumed by the transition are fed into the corresponding LAMPS graph, and the output tokens of the LAMPS graph are then put into the output places of the transition. This way, on an abstract level a LAMPS graph \((P \cup T, R)\) with input places \(I\) and output places \(O\) can be viewed as a transaction with the signature \(\text{sig}(t) = (I, O)\). This allows to view a process or behaviour on various levels of abstraction. Figure 1 depicts the hierarchical use of abstractions.

**GENETIC ALGORITHM**

This section gives a short introduction to genetic algorithms and then describes the particular genetic operators that have been developed for LAMPS graphs.

**Genetic algorithms in general**

Basically, the task of the learning algorithm is to put together LAMPS building blocks - i.e. transactions and places - in an appropriate order. The set of primitive transactions is defined by the actions that the agents can execute in the given application. The set of places can grow arbitrarily during the learning process.

Genetic algorithms are well-suited for this learning problem. The general idea behind genetic algorithms is to mimic natural evolution (Goldberg and Holland 1988). A population of solutions is evaluated with regard to a fitness function. The fitness function may measure the degree to which the learning target is fulfilled. The population goes through a series of generations over time. The algorithm starts with a generation of randomly constructed strategies. Each generation evolves from the previous one under the influence of genetic operators. There are the following operators:

- **Cross-over**: This operator mimics the reproduction of two parent solutions into a new child solution. Conceptually, the child should contain traits or parts of both parents, while forming a new solution.

- **Mutation**: The mutation operator is applied randomly to some solutions and modifies them in a random fashion.

- **Selection**: The selection operator decides which solutions are selected for cross-over.

- **Replacement**: This operator selects the solutions that are transferred into the next generation.

The genetic algorithms on their own are bias-free. The learning bias is introduced by a) the fitness function and b) by the representation language for the solutions. In this paper, the fitness function is specified by the application. Thus, we chose to use the building blocks of LAMPS for defining the learning bias. However, another challenge is the representation of the solutions for the genetic operators. Traditionally, solutions are represented as bitstrings, since these can be processed efficiently by the genetic operators. However, mapping LAMPS graphs to bitstrings is a challenge,
particularly if one requires that the operators produce only bitstrings that correspond to syntactical correct LAMPS graphs. For this reason, we decided to apply the genetic operators directly on the LAMPS graph representations. To this end, we introduce a mutation and a cross-over operator that process LAMPS graphs.

Cross-over

The requirements for a cross-over operator that generates a new LAMPS graph from two parent graphs are the following:

- **Correctness of the resulting LAMPS graph:** The resulting graph must obey the constraints as required by LAMPS. That is, the graph must be bipartit and the rules of transitions must be syntactically correct. Furthermore, the types of input and output places must match the signature of connected transitions.

- **One connected graph:** For the application it is important that the result of the cross-over does not consist of unconnected subgraphs, but is one connected graph.

- **Coverage of both parents:** The resulting graph should contain parts of both parents.

- **Explorative nature:** The new solution should be sufficiently distinct from the parent solutions in order to guarantee a good exploration of the solution space.

In order to meet these requirements, we based the cross-over operator on an existing graph-matching algorithm that computes the intersection of two graphs (Bengoetxea 2003). Conceptually, in the first step the intersection of both parent graphs is identified. If the intersection is empty, then no child graph is produced. In the second phase the intersection graph is extended by one transaction of each parent that is connected to the intersection but is not part of the intersection. If such a transaction does not exist, the intersection is not extended.

Figure 2 shows an example for the cross-over operation. This approach fulfills the requirement for correctness of the resulting graphs. Since all operations process subgraphs no modification is done that could violate the correctness axioms. Due to the use of the intersection of both graphs and connected transactions it is guaranteed that the resulting graph is connected and that parts of both parents are used. Furthermore, since those parts that do not belong to the extended intersection are discarded, and since the intersection is extended with different parts of the parents, the resulting graph is sufficiently different from the parents. One can construct examples in which the child graph is identical to one or even both of the parents, but in an average case analysis this does not happen often. Especially for large and complex graphs, the cross-over results in child graphs that differ from both parents.

Mutation

For the mutation operator the same requirements hold as for the cross-over operator, with the exception of the covering of both parents.

With the idea of LAMPS building blocks in mind, the mutation operator was designed to allow two modifications only. One modification is to delete a random transaction. Since the resulting graph must be still connected, any places that are isolated after deletion of the transaction are discarded. If the graph is split into two or more subgraphs after the deletion, then the mutation operator is not applied to that transaction.

The other modification is to add a random transaction to random, yet fitting places. That is, the places must be of the type that is required by the transaction. With a certain probability new places are added as input or output places to the new transaction. However, at least one existing place must be connected to the transaction in order to guarantee the connectedness of the graph.

Thus, replacement of a transaction by another transaction must thus be generated by a series of deletion- and add-mutations.
Selection and replacement

Standard methods for selection and replacement have been applied. For selection, Stochastic Universal Sampling (Baker 1987) was chosen, since this relatively simple method leads to good diversity in the population. Overhasty convergence to a relatively good (relative wrt. rest of the population) solution is avoided in order to prevent reaching local optima.

For replacement, the method "generation exchange" (Goldberg and Holland 1988) was chosen in order to avoid local optima. It guarantees that the average quality of a generation is at least as good as the quality of the predecessor generation.

EXPERIMENTS

The approach has been evaluated in the domain of medical evacuation. The setting is that injured people are distributed over a specified region with the size of a medium-city (with a population of approx. 350,000). There are different levels of injury, ranging from light to severely wounded. The latter have to be evacuated fast, otherwise they perish. With each time step, the health points of the injured agents are reduced by a certain amount. The so-called blue team has available rescue helicopters and several emergency vans. The goal of the simulation is to evacuate as many injured people to the lazarett as possible.

Each simulation run contained 10 heavily injured, 10 medium injured and 10 lightly injured agents. One helicopter and two vans were available to collect and deliver them to the lazarett.

The genetic algorithm was run with the following parameters. The population size (that is, the number of strategies that were evaluated against each other), was 100. The fitness function that was to be maximized summed up the health-points of all injured agents at the end of the simulation run. A simulation run was complete after all injured agents were either delivered to the hospital or had perished. Note that the time needed for evacuation was not directly included in the fitness function. Rather, with time, the health points of the injured agents diminished. Thus, a faster rescue strategy received higher fitness.

Figure 3 depicts the fitness function over the generations. After already 51 generations, the learnt strategies are nearly optimal. An analytical optimum is hard to calculate, since the dynamics of the setting are complex. However, at 1500 health points, the health of the injured agents is only 100 points below the starting condition. Thus, the learnt strategy delivered the agents quickly, so that only few health points were reduced during time. Apart from this quantitative criteria, the qualitative analysis of the learnt strategies revealed convincing results, too. The strategies that were evaluated during the 65 generations covered a wide range of possibilities.

Figure 3: Plot of a learning run. The fitness value is depicted on the vertical axis, the generations on the horizontal axis.

However, most of them were pruned during evaluation, so that among this variety of strategies only the best strategies survived. The best strategies were similar in their overall aspects: Without domain knowledge, the genetic algorithm learnt to use helicopters for the heavily injured agents, and the slower vans for lightly injured agents. While this is an intuitive strategy, the genetic algorithm acquired this without human intervention.

CONCLUSIONS AND OUTLOOK

This paper presented a machine learning approach for automatically acquiring agent-behaviors in agent-based simulation. We showed that the petri-net-based language LAMPS is a good representation language to learn agent-behaviors. The learning algorithm requires as input a set of domain-specific actions and returns an agent-behavior represented as LAMPS graphs. It was shown experimentally that the approach can acquire working agent behaviors and that the learnt behaviors are nearly optimal.

The approach can substantially increase the reliability of agent-based simulations. While in this work, the agent behaviors were learnt from scratch, it seems feasible that the learning approach can also be used to optimize strategies that were initially hand-coded by human modellers. This can be achieved by starting with a population of hand-coded strategies rather than randomly constructed strategies.

In the long run, it might even be possible that the automatic learning of agent-behaviors will replace manual work altogether. This will require further research in the area of behavior-representation, machine learning and validation.

ACKNOWLEDGEMENTS

Part of this work has been conducted under contract for the department A5 of the IT-office of the German armed forces.
forces (IT-AmtBw). At this institution, the project is overseen by Major T. Doll, whose valuable contributions are gratefully acknowledged.

REFERENCES


AGENT
BASED
MANAGEMENT
Quality of Service in Healthcare Units

José Machado1, António Abelha 1, Paulo Novais 1, João Neves 2 and José Neves 1
1Universidade do Minho, Departamento de Informática, Braga, Portugal
jmac@di.uminho.pt abelha@di.uminho.pt pjon@di.uminho.pt jneves@di.uminho.pt
2Centro Hospitalar de Vila Nova de Gaia e Espinho, EPE, Portugal
j.neves@hotmail.com

ABSTRACT

Healthcare systems have to be understood in terms of a wide variety of heterogeneous, distributed and ubiquitous systems, speaking different languages, integrating medical equipment and being customized by different companies, which in turn were developed by people aiming at different goals. An architecture has been envisaged to support the medical applications in terms of an agency for integration, diffusion and archiving of medical information and the Electronic Medical Record, a form of a web spider of an intelligent information processing system, its major subsystems, their functional roles, and the flow of information and control among them, with adjustable autonomy. With such web based simulated systems, quality of service will be improved (e.g., the available knowledge may be used for educational and training purposes).

INTRODUCTION

By definition, following [1], an agent is a self-contained, interactive and concurrently-executing object, possessing internal state and communication capability. On the other hand, health care systems have been for some years a very attractive domain for Computer Science researchers, considering the great potential for information integration and automation, and this is an issue of study in which Medicine and Agent based technologies and methodologies for problem solving may overlap.

We want to find out how usability is maintained in this new landscape of social and ubiquitous computing, when applied to health care units, a domain that presents the strongest social focus, and thus offers the biggest challenge for socially aware software systems design [4] [3]. We will describe how the ubiquitous computing system has become an everyday affair and how the traditional usability methods are inadequate to guide the design of such systems, that are increasingly social, both in its virtual realm and in its context of use.

This makes health care a perfect target for the application of autonomous and intelligent agents, since conventional systems are naturally bounded by their lack of rich knowledge representation, integration and communication. Health care systems have to be addressed in terms of a wide variety of heterogeneous, distributed, and ubiquitous systems speaking different languages, integrating medical equipment and customized by several companies, which in turn were developed by people aiming at different goals.

This lead us to consider the solution to a particular problem, to be part of an integration process of different sources of information, using different protocols, in terms of an Agency for the Integration, Diffusion and Archive (AIDA) of medical information, and the Electronic Medical Record (EMR) software (in Portuguese referred as PCE - Processo Clinico Electrónico), bringing to the health care arena new methodologies for problem solving, computational models, technologies and tools, and enabling ambient intelligence or ubiquitous computing [12]. With access granted to Clinical and Historical Databases, agent technology may provide answers to those who give assistance to patients with a maximum of quality and medical evidence [14].

AGENT ORIENTED PROGRAMMING

Although there is no universally accepted definition of agent, in this work such an entity is to be understood as a computing artefact, being used in hardware or software devices, that exhibit the properties:

- autonomy; i.e., whereby such entities have the ability to act without the direct intervention of their peers, namely humans;

- reactivity; i.e., whereby such entities are situated in an environment that can perceive through sensors and act in reaction to stimuli (e.g., revising their beliefs according to or in reaction to new inputs);

- pro-activity; i.e., whereby such entities exhibit intelligent problem solving capabilities (e.g., planning their activities in order to achieve short or long term goals); and

- social behaviour; i.e., whereby such entities are aware of one another, can interact with one another and may modify their behaviour in response to others; can communicate via a set of low or high level constructs and protocols as well as means of addressing and direct communication; can cooperate
in order to achieve joint as well as individual goals, what means that must have the ability to negotiate with other agents either to accomplish their own goals or to joint plans to achieve common goals; to perform belief revision in the context of additional sources of information provided by their peers.

To develop such systems, a standard specification method is required, and it is believed that one of the keywords for its wide acceptance is simplicity. Indeed, the use of intelligent agents to simulate human decision making in the medical arena offers the potential to set an appropriate software development and analysis practice and design methodology that do not distinguish between agent and human, until implementation.

Being pushed in this way, the design process, the construction of such systems, in which humans and agents can be inter-changed, is simplified; i.e., the modification and development in a constructive way, of multi-agent healthcare systems with a human-in-the-loop potential aptitude is becoming central in the process of agent-oriented software development and analysis.

These systems have provided a clear means of monitoring the agents’ behaviour with significant impact in their process of knowledge acquisition and validation. Multi-Agent Systems are a natural connection to intelligent systems evolution, being elements for task substitution or delegation, usually performed by human beings. However agent based systems have some restrictions, such as global system control and universal view absence, and some want of confidence and fear of competence delegation by human beings. To delegate tasks, bilateral confidence relations have to be established. Organizations may also mature their experience relatively to the use of autonomous software components.

The traditional programming languages do not support the description of certain types of behaviour which usually involves computational agents [17]. In genesis, systems that incorporate those functionalities have a multi-layer architecture, evolve from esoteric software subsystems, network protocols, and the like.

An agent must be able to manage its knowledge, beliefs, desires, intentions, goals and values [8]. It may be able also to plan, receive information or instructions, or react to environment stimulus. It may communicate with others agents, share knowledge and beliefs, and respond to other agents upon request. It may cooperate for diagnosing errors or information faults in its knowledge bases, sharing resources, avoiding undesirable interferences or joining efforts in order to revisit the knowledge bases of its own and of its peers, in order to reach common goals [12]. Agents exchange messages which are well-formed formulae of the communication language, performing acts or communicative actions [9].

Agents in a health care facility configure applications or utilities that collect information about the assets in the organization [7]. Once that information has been collected it can be posted directly to other entities (e.g., a physician), or a server, saved in a file or emailed to someone to be handled at a later date, or sent using HL7 (Health Level 7) (http://www.hl7.org) or web services. HL7 plays an essential role in extending the interoperability for the development of health information exchange, in the standardization of XML medical document structures and in the specification of robust vocabulary definitions for use in clinical messages and documents (e.g., SNOMED CT) (http://www.hl7.org) enabling functional specifications for the EMR. HL7 is an international community of healthcare subject matter experts and information scientists collaborating to create standards for the exchange, management and integration of electronic healthcare information. HL7 promotes the use of such standards within and among healthcare organizations to increase the effectiveness and efficiency of healthcare delivery for the benefit of all the human beings. The following text is an example of a HL7 message:

```
MSH|&amp;IS|AIS|EMR|UM|20060928104242|1
ADT“A04|A200609281042325331|P|2.4||AL
EWN|A04|1111
PID|&|2320464|SH|NS|
170445890|SH|SSN|GOMES“PAULO”||
19800112|M|1|CASA SEIA REAL”~”BRAGA”
4700|253811072|11111|0|SH|
PV1|1|INT|1|26|1|1111111111111|6198025|SH|
IN1|1|935601|SH|935601
11111111111111111111111111111
11111111|1704458909|
IN2|1|1|11111111111111111111111111111
11111111111111111111111111111
```

An acknowledgement message is received later with a successful state:

```
MSH|&amp;IS|AIS|EMR|UM|20060928104242|1
ACK|AA20060928104232549|P|2.4
MSA|AA|20060928104232549|
```

Indeed, in a hospital, the collection of vast amounts of medical data will not only support the requirements of archiving, but also provide a platform for the application of data mining and knowledge discovery techniques to determine possible medical trends and the real data to support educational training. Knowledge discovery techniques can be applied to identify pathologies and disease trends.

The data can also be used for educational and training purposes; unique cases can be identified and used to advise practitioners. This may lead to the goal of Ambient Intelligence at Health Care Units at our doorstep [6] [5]. The interfaces are based on Web-related front-ends, querying or managing datawarehouses. Such an approach can provide decision support. A context dependent formalism has been used to specify the AIDA
system incorporating facilities such as abstraction, encapsulation and hierarchy, in order to define:

- the system components or agents;
- the socialization process, at the agent level and the multi-agent level, following other possible way of aggregation and cooperation;
- the coordination procedure at the agent level; and
- the global system behaviour.

On the other hand, Information sources in a healthcare unit:

- are distributed, heterogeneous, large and complex;
- integrate medical equipments that speak different languages; and
- are built around information systems customized by several companies using different operating systems, languages, applications and hardware.

Communications are sometimes limited by old infrastructures and new projects collide with financial restrictions and bureaucratic delays. The homogeneity of clinical, medical and administrative systems is not possible due to financial and technical restrictions, as well as functional needs. The solution is to integrate, diffuse and archive this information under a dynamic framework, in order to share this knowledge with every information system that needs it. AIDA, as an agency that provides intelligent electronic workers, here called pro-active agents, and in charge of tasks such as communicating with the heterogeneous systems, sends and receives information (e.g. medical or clinical reports, images, collections of data, prescriptions), managing and saving the information and answering to information requests, with the necessary resources to their correct and in time-line accomplishment.

AIDA also supports Web based services to facilitate the direct access to the information and communication facilities set by the humans, i.e. AIDA construction follows the acceptance of simplicity, the conference of the achievement of common goals and the addressing of responsibilities.

The main goal is to integrate, diffuse and archive large sets of information from heterogeneous sources (departments, services, units, computers, medical equipments); AIDA also provides tools in order to implement communication with human agents based on web based services. Under these presuppositions, a Healthcare Information System (HIS) will be addressed in terms of (Figure 1):

- The Administrative Information System (AIS), which intends to represent, manage and archive the administrative information during the episode (an episode is a collection of all the operations assigned to the patient since the beginning of the treatment until the end);
- The Medical Support Information System (MIS), which intends to represent, manage and archive the clinical information during the episode;
- The Nursing Support Information System (NIS), which intends to represent, manage and archive the nursing information during the episode;
- The Electronic Medical Record Information System (EMR);
- The Information Systems (DIS) of all the departments or services, in particular of the laboratories (Labs), Radiological Information System (RIS) and Medical Imaging (PACS - Picture Archive and Communication System), which deals with images in a standard format, the DICOM one.

An architecture has been envisaged to support the medical applications in terms of AIDA and EMR, a form of a web spider of an intelligent information processing system, its major subsystems, their functional roles, and the flow of information and control among them, with adjustable autonomy. Indeed, many complex systems are made up of specialized subsystems which are understood as intelligent agents that interact in flexible, goal-directed manners, and are understood as theories; i.e., the intelligence of such a system as a whole arises from the interactions among all the system entities. Such systems are therefore a purely communicative Multi-Agent System (MAS), i.e., there is no external environment influence and the agents communicate only through messages. The general architecture of such systems is given in Figure 2, being a brief description of the different types of agents involved given below, which, in turn, will allow one to measure or quantify the quality-of-service (qos), a subject that will be addressed in the section about the Computational Model:
Intuitively, not \( p \) is true whenever there is no reason to believe \( p \), whereas \( \neg p \) requires a proof of the negated literal. An extended logic program (program, for short) is a finite collection of rules and integrity constraints, standing for all their ground instances, and is given in the form:

\[ p \leftarrow p_1 \land \ldots \land p_n \land \text{not}q_1 \land \ldots \land \text{not}q_m; \quad \text{and} \]

\[ \text{?}p_1 \land \ldots \land p_n \land \text{not}q_1 \land \ldots \land \text{not}q_m, \quad (n, m \geq 0) \]

where \( \text{?} \) is a domain atom denoting falsity, the \( p_i \), \( q_j \), and \( p \) are classical ground literals, i.e. either positive atoms or atoms preceded by the classical negation sign \( \neg \) [21]. Every program is associated with a set of abducibles. Abducibles can be seen as hypotheses that provide possible solutions or explanations of given queries, being given here in the form of exceptions to the extensions of the predicates that make the program. These extended logic programs or theories stand for the agents (or programs) that populate the universe of discourse. Indeed, in our approach to AOP, we will not get a logic representation of the agents that model a particular universe of discourse, but rather a possible representation of such agents which may be optimized. On the other hand, logic programming enables an evolving agent to predict in advance its possible future states and to make a preference. This computational paradigm is particularly advantageous since it can be used in program synthesis, employing the methodologies for problem solving that benefit from abducibles, in order to make and preserve abductive hypotheses [22] [23]. Indeed, it is on the preservation of the abductive hypotheses that our approach will be based to present a solution to the problem of agent or program optimization and evolution.

Designing such a selection regime presents, still, unique challenges. Most evolutionary computation problems are well defined, and quantitative comparisons of performance among the competing individuals are straightforward. By contrast, in selecting an abstract and general logical representation or program, performance metrics are clearly more difficult to devise. Agents (or programs) must be tested on their ability to adapt to changing environments, to make deductions and draw inferences, and to choose the most appropriate course of action from a wide range of alternatives.

Above all they must learn how to do these things on their own, not by implementing specific instructions given to them by a programmer, but by continuously responding to positive and negative environmental feedback.

In order to accomplish such goal, i.e., to model the universe of discourse in a changing environment, the breeding and executable computer programs (or agents) will be ordered in terms of the quality-of-information that stems out of them, when subject to a process of conceptual blending [24].

In blending, the structure or extension of two or more predicates is projected to a separate blended space,
which inherits a partial structure from the inputs, and has an emergent structure of its own. Meaning is not compositional in the usual sense, and blending operates to produce understandings of composite functions or predicates, the conceptual domain, i.e., a conceptual domain has a basic structure of entities and relations at a high level of generality (e.g., the conceptual domain for journey has roles for traveler, path, origin, destination). In our work we will follow the normal view of conceptual metaphor, i.e., metaphor will carry structure from one conceptual domain (the source) to another (the target) directly (Figure 3). In Figure 3 INPUT denote the agents (or programs) that are object of optimization and correlative evolution.

We construct a dynamic virtual world of complex and interacting population of agents, entities that are built as evolutionary logic programs or theories that compete against one another in a rigorous selection regime, in order to produce the optimal model to a particular problem. In other words, the agents or logical theories evolve in order to model the universe of discourse, in which fitness is judged by one criterion alone, the quality-of-information [21].

It is now possible to measure or quantify the quality of service, which is given in terms of the logic program or theory, defined at meta-level, according to the logic program (or agent) given below:

\[
\neg qos(Agent,Q_{service}) \land \\
\neg exception_{pa}(Agent,Q_{service}), \\
exception_{pa}(ag_{pa}, 0.5), \\
exception_{pa}(ag_{pa}, 0.33), \\
\neg ((pa(ag_{pa}, X)) \lor \\
\neg pa(ag_{pa}, Y)) \land \\
\neg ((pa(ag_{pa}, X)) \land \\
\neg pa(ag_{pa}, Y))), \\
exception_{pa}(X, Y) \leftarrow da(ag_{pa}, Y),
\]

where the integrity constraint or invariant for \( ag_{pa} \) stated above denotes an exclusive or, i.e., the quality of information associated with the \( ag_{pa} \) is tailored by the exceptions referred to above for \( ag_{pa} \) (in this case the value of 0.5). Therefore, the quality of information reported by the qosagent for the Healthcare Information System is given the dashed area of the Figure 4.

**TOWARD VIRTUAL HOSPITALS**

Virtual hospital allows healthcare workers and professionals to undertake simulated medical scenarios for patient care, providing a projection of a real hospital, using computerised tools to resolve health problems from physical attributes, and simulating conversational dialogue in the area of Medicine. The virtual hospital offers a wide digital library of cases under supervision allowing the attendance of medical and nursing education courses, integrating highly heterogeneous sources of information into AIDA.

The system content is created automatically by the physicians as their daily work goes on and students are encouraged to articulate lengthier answers that exhibit deep reasoning, rather than to deliver straight tips of shallow knowledge. The goal is to take advantage of the normal functioning of health care units to build on the fly a knowledge base of cases and data for teaching and research purposes. Medical simulation facilitates the widespread acceptance of new techniques in health units and allows the participants to practice diagnosis,
medical management and behavioural approaches, with no risk to patient safety. Simulation is a dynamic artificial environment where conditions or hypotheses are created in order to study or experience real facts or activities. Simulation is very popular in the area of Education and Training. In particular, simulation of real-time environments has been focused as a tool for teaching in many areas, especially in medicine where pedagogical practices carry out high risks and costs, in professional and law points of view, enabling the creation of artificial environments for pedagogical purposes. Students can learn making mistakes in safe environments. Simulation researchers need to focus more on developing practical methods for building simulations that use the web and producing quality simulation software and flexible services. Those goals have been focused in AIDA and EMR (Figure 5).

For example, despite the advancements in the area of the health care, every year around one hundred thousand patients contract an infection at the hospital environment and more than ten thousand die in consequence of these infections. Beyond the morbidity and considerable mortality, the hospital infection provokes a considerable increase of costs with trials of diagnosis, of therapy and with the increase of the time of hospital admission. The hospital or nosocomial infection is an infection that is not present at the moment of patient admission to the hospital. The registration of information about this phenomenon in databases is, more and more, a reality, turning viable the representation of this information through mathematical formalisms that, conjugated with the application of Artificial Intelligence techniques, allows for the discovery of knowledge related with the critical factors, processes and infectious items. The ultimate goal has been to construct a web-based computational tool to automate the registration process to support the clinical body work, monitoring the performance and identifying the procedures through intelligent agents from AIDA, implemented in order to reduce the impact of the infections.

This system is also now a reality in Hospital Geral de Santo António, in the Intensive Care Unit, being developed and configured from generated forms of the EMR and sharing information through AIDA [2]. Another interesting application of ubiquitous intelligent systems is described in [6]. EMR and AIDA are the starting point for the creation of ambient assisted living ubiquitous environment for improving the quality of life of the elderly.

CONCLUSIONS

We intend to specify, to develop and to implement data processing computational systems starting from an online structured and systematized recording of information, enabling the recording of diagnosis, the register of the flat therapeutic being prescribed, the cooperation between the laboratory and the clinical ones, as well as the implementation of politics of prevention. Indeed, we are looking at software programs to improve and to measure the quality of service, using autonomous and intelligent agents.

Although this may seem the description of search methods in general, it is clear that we may deal with wider amplitudes of divergence in tasks that demand higher creativity. These tasks do not necessarily have to present a particular form or be of a specific kind. However, the quest for a previously unseen and correct solution is surely expected. A solution that traditional methods do not seem to find. We think that some qualitative jump must be made, such that classical methods become more able to diverge or at least to combine with other processes.

ACKNOWLEDGEMENTS

We are thankful to the Hospital Geral de Santo António (HGSA), in Oporto, Portugal, for their help to the analysis and development of the AIDA and EMR systems referred to above, which are now being largely used in three important health care units in Portugal.

REFERENCES


[16] L. Weed, Medical records, medical education, and patient care. The problem-oriented record as a basic tool, Case Western Reserve University, Cleveland, OH, 1969.


ENTERPRISE KNOWLEDGE MANAGEMENT SUPPORTED BY
AGENT BASED MODELING AND SIMULATION

Dr. Marco Remondino
Computer Science Department
University of Turin
Tel. +39-011-6706711
remond@di.unito.it

Prof. Marco Pironti
Computer Science Department
University of Turin
Tel. +39-011-6706711
pironti@di.unito.it

1 Abstract

In this work we propose agent based simulation as a paradigm able to capture the emergent complexity typical of a knowledge base inside an enterprise. In general knowledge coincides with the world of the economic relations and the technologies, and is therefore made tangibly enjoyable through the intervention of the markets and of science. An enterprise is an Economic system and is complex by nature; in previous works it has been shown how agent based modelling can be a viable way to represent and capture emergent knowledge, sometimes referred to as “implicit”, in order to disambiguate it from the “explicit” one. Different agent based paradigms are reviewed and a CMS is described in detail, showing it’s a complex system and hence can be modelled using the agent based approach. In future works we’ll use UML formalisms to project an object oriented model using agents.

2 Introduction

The classical approach views knowledge as an objective resource; use availability and transparency makes it available to everyone. Knowledge coincides with the world of the economic relations and the technologies, and is therefore made tangibly enjoyable through the intervention of the markets and of science. In a system characterised by economic relations and by technology, knowledge is on the one hand a fundamental and non-pervading resource but, on the other hand, it is so much taken for granted that it is not regarded as a source of competitive advantage. The obviously problematic nature of such assumptions caused Herbert Simon to take a major step: he identified the knowledge “problem”, which in this way became known in the studies concerning organisation with the term “rationality”. By creating the term “classical” or “absolute rationality”, versus the term “bounded rationality” (Simon 1972) or “procedural rationality” (Simon 1976) theorised by him, he stated that a classical rationality exists, that it is one of the forms and not “the” only form of rationality, and that, as such, it can be subject to criticism. In particular, it is stressed that all human (and artificial) agents have a limited ability to process information, that acquiring information involves a cost and that hence the information available for decision-making purposes is necessarily limited. As a result of these assumptions, people make their decisions with a limited knowledge of the situation, by employing routines which are meant as experiential processes and by adopting information technologies to overcome rationality limitations by means of increased information storage and processing capacity.

A second trend, of a cognitive nature, is related to the classical one thanks to March’s critical contribution (1991), stressing how the rationality problem does not lie so much in the limited amount of information available as in the fact that the information changes depending on the context and may therefore be interpreted rather than simply being acknowledged. The problem shifts from the requirement to select information, as suggested by Simon, to that of interpreting it, as suggested by March.

In terms of Knowledge Management (KM), the rationalist approach, owing to its being rooted in western culture and in the traditional managerial models (Nonaka 1995), is the one which has mostly inspired the managerial solutions and literature on the subject. On the one hand, the non-problematic nature of information, meant as a vehicle of content irrespective of the subjectivity of senders and recipients, leads to conceive the KM systems in terms of knowledge markets (Davenport, Prusak 2000), populated by knowledge producers, intermediaries and consumers. The purpose of a KM system is therefore that of demolishing the barriers which prevent this type of market from becoming efficient, by increasing information completeness (making the real value of knowledge explicit), overcoming unevenness (some have greater access to knowledge) and avoiding local interests (knowledge only exchanged among parties that are close to each other). The overall complexity behind the generation and transmission of knowledge and hence behind its management, make it a perfect field to investigate and study through agent based techniques, in particular agent based modelling and simulation. According to (Ostrom 1988), simulation can be considered a third way to represent social models; in particular, it can be a powerful alternative to other two symbol systems, the verbal argumentation and the mathematical one. Simulation has a great advantage over the other two, which is to be found in
its high portability on a computer, through a program or a particular tool. Computer programs can then be used to model either quantitative theories or qualitative ones. In a social context, the single parts and the whole are often very hard to describe in detail. There are 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 often complex and difficult to predict, and which can be used in open and distributed systems. A software agent can be described as a flexible system, capable of dynamic, autonomous actions, in order to meet its design objectives, that is situated in some environment. The main features for a software agent are: situatedness, that is ability to perform actions according to a particular input received from outside, which can, in turn, change the environment itself; autonomy in performing actions, without intervention of humans; flexibility and adaptability. Some particular agents can also be proactive, which means they are goal-directed, and social, in the way they can interact with other artificial agents, robots, and humans. Such an intelligent agent can be referred to as a Belief-Desire-Intention (BDI) one. The goal of this paper is to show how agent based models can capture the emerging behaviour of complex social systems and hence can represent, with a structured framework, the role of explicit and implicit knowledge within organizations.

3 A Taxonomy for Agents

Agents themselves have traditionally been categorized into one of the following types (Woolridge and Jennings, 1995): Reactive, Deliberative (Cognitive), Hybrid. When designing any agent based (AB) system, it's 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 (aka cognitive) 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 which bases its actions on the predicted actions of other agents is deliberative. In Mataric (1995) we read that reactive agents maintain no internal model of how to predict future states of the world. They choose 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. These types of agents are sufficient for limited environments where every possible situation can be mapped to an action or set of actions. The purely reactive agent's major drawback is its lack of adaptability. This type of agent cannot generate an appropriate plan if the current world state was not considered a priori. Different from reactive agents are the deliberative ones. The key component of a deliberative agent is a central reasoning system (Ginsberg, 1989) that constitutes the intelligence of the agent. Deliberative agents generate plans to accomplish their goals. A world model may be used in a deliberative agent, increasing the agent's ability to generate a plan that is successful in achieving its goals even in unforeseen situations. This ability to adapt is desirable in a dynamic environment. The main problem with a purely deliberative agent when dealing with real-time systems is reaction time. For simple, well known situations, reasoning may not be required at all. In some real-time domains, such as robotic soccer, minimizing the latency between changes in world state and reactions is important. Hybrid agents, when designed correctly, use both approaches to get the best properties of each (Bensaid and Mathieu, 1997). Specifically, hybrid agents aim to have the quick response time of reactive agents for well known situations, yet also have the ability to generate new plans for unforeseen situations.

4 Multi Agent Systems (MAS)

A multi agent system can be thought of as a group of interacting agents working together to achieve a set of goals. 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.

5 Modeling and Computational Simulation

Modeling is a way of solving problems that occur in the real world. It is applied when prototyping or experimenting with the real system is expensive or impossible. Modeling allows to optimize systems prior to implementation. It includes the process of mapping the problem from the real world to its model in the world of models, – the process of abstraction, – model analysis and optimization, and mapping the solution back to the real system. We can distinguish between analytical and simulation models. In analytical, or static, model the result functionally depends on the input. However, analytical solution does not always exist, or may be very hard to find. Then simulation, or dynamic, modeling may be applied. A simulation model may be considered as a set of rules (e.g. equations, flowcharts, state machines, cellular automata) that define how the system being modeled will change in the future, given its present state. Simulation is
the process of model “execution” that takes the model through (discrete or continuous) state changes over time. In general, for complex problems where time dynamics is important, simulation modeling is a better answer.

6 A CMS Overview
A Content Management System (CMS) is responsible for gathering, managing and publishing blocks of information which are regarded as content. Furthermore, it allows, within certain contexts, to carry out analyses on the type of use to be made by recipients. By going through the process shown in the layout (Fig. 2), it is possible to understand, moving from the top to the bottom, all the process that a piece of information has to go through in order to be converted into content. In short, first we have the gathering from different sources, with a conversion into a standard format and enrichment as far as information is concerned, which enables the material to start looking like content. After this, the information is stored in a centralised file, where it is subject to evaluation and review, and, thanks to automated procedures, distributed to distinct channels. Furthermore, a reverse flow can result from the data entry relating to the content utilisation data, which become available as new information. It should be noted that, even if logically separate, the four chief flows may also happen to overlap each other.

7 The Gathering System
A CMS gathering system comprises the processes aimed at the proper arrangement of the information in order for it to become content. This first stage includes the following processes:
- creation;
- acquisition;
- conversion;
- aggregation;
- gathering.

7.1 Creation
Creation relates to the content production process. It differs from the acquisition process owing to the human intervention of an author. If the content is created for purposes other than CMS-related aims, then we are referring to acquired content.

7.2 Acquisition
Acquisition is a gathering process involving information which was not originally created for CMS. This process can be partially or totally automatic.

Acquirable information chiefly comes from three types of sources:
1) Syndication: the term syndication relates to the services dedicated to content sale and supply. We are therefore referring to information which is by nature aimed at reutilisation: this means that the information comes in a “convenient” digital format and, above all, it is already segmented and provided with meaningful meta-data.
2) Information systems: many of the data representing the content base are usually already present within a company’s information system. Data coming from administration management, from ERP, CRM systems, etc. can be gathered and integrated as part of the CMS in order to be eventually processed and published.
3) Documents and files: we can include as part of this category any pre-existing information stored in electronic format. As a rule, this type of file is not meant to be re-used and it is not therefore characterised by standard formats or metadata. For this reason its inclusion in a CMS requires a significant work at a conversion and aggregation system level.

7.3 Conversion
Once gathered through an Information Gateway (information access point), the information is ready to be dealt with in order for the CMS systems to easily manage them. The first process it has to go through is the conversion process, required when the acquired information is not available in the format and structure requested by the system. The conversion process is characterised by the following three logical steps:  
- Refinement: this step is required for the removal of accessory information which is not needed.
- Format conversion: the original digital format in which the information is presented is converted into a format supported by CMS.
- Structure conversion: in this phase, the structure is confirmed or modified if required. The process results in the achievement of information that is as in line as possible with the standards used to structure the data contained in a system.

4.4 Aggregation
Aggregation is the process required to classify information. Thanks to this stage, incoming information is

---

1 Information bearing the data that the computer employs to organise and arrange the gathering, management and distribution.
2 ERP: Enterprise Resource Planning, work planning systems; CRM: Customer Relationship Management, customer relation strategy systems
analysed and broken down into elements, each of which is then coupled with descriptive data. This is the key step required to convert information into content. Aggregation processes can take place in two different ways or modes:

- Manual Aggregation: most part of the information has to be subject to human supervision in order to define classification criteria.
- Automatic Aggregation: for certain information categories, the process can be managed by artificial agents capable of recognising its content and automatically link it to the meta-data.

Aggregation is based on a series of predefined taxonomies used to classify the information within a structure or within recognisable categories. For complex content items, only human intervention is capable of properly selecting adequate information. Data processing procedures, on the other hand, can evaluate information content and, based on preset rules, decide how they ought to be classified. Furthermore, certain intelligent systems are capable of self-learning classification methods based on experience.

7.5 Gathering
The gathering process is the final step of this first stage. It includes all the necessary tools and procedures required (whether manual and/or automatic) for entering the content items processed by the central file, where the management system is based. The most common method whereby authors create content directly on the CMS are web-based interfaces. In these forms, the authors enter blocks of text, meta-data and images or other media. After this, the form content is processed and stored on file. In addition, the content can be created through application packages external to the CMS and then uploaded onto the system once complete. Even though web forms make controls and content confirmation easier, word processing tools certainly offer additional flexibility. One of the automatic gathering processes is the one taking place through the Feedback Gateway, through which it is possible to use data coming from analysis processes. These are data deriving from the recording of users’ behaviour and from reports coming from those who have benefited from the published content. Thanks to its predefined format and structure, this information does not require conversion or aggregation.

8 The Management System
The management system in a CMS is aimed at a long-term storage of content and all other processes which content is subject to before being published. The management system is capable of communicating information such as:

- Details on content items and on the life cycle phase they finds themselves in.
- The status of working staff and any bottlenecks.
- Published content and items which should eventually be removed.

The management system comprises:

a) The Filing System
The central file is made up of the databases and file folders storing all the content and data required for operating the CMS.

The content items can be stored inside a CMS as databases or in file format. The databases may be of a relational type (RDBMS) or based on objects (such as XML). A relational database employs tables, lines and columns to present the content; in an object database, the content items are directly expressed in an XML format and stored in a hierarchical sequence. These two technologies do not mutually exclude each other; in fact it is possible to find relational database systems which store XML-structured data within tables.

b) Workflow
The workflow is made up of the processes which enable monitoring of content processing for each stage. It facilitates coordination and programming of the working activities of staff employing the CMS.

The system is on the whole based on preset rules accompanying content processing, and on user profile allocation policies involving differing user rights to accomplish certain activities.

In general terms, CMS users are divided into three categories:

- Editors: staff entering the content;
- Publishers: staff publishing the content;
- Supervisors: staff supervising editors and publishers.

Of course the workflow system described above is extremely simplified; in real life, the process can be definitely more complex, be made up of several steps and provide not only for serial approvals, but also for parallel approvals, in which each content item, in order to be admitted to the next stage, needs to be confirmed by several people.

c) Processing
Certain types of data can prove difficult to read for those consulting published content; for this reason, it may prove useful to integrate in a CMS a processing system capable of transforming such data in graphs or tables, for instance, so as to ensure a more effective communication process.

9 The Publishing System
The publishing system handles content extraction from the central file and its publication. It is chiefly made up by the following three elements:

a) template system;

3 RDBMS: Relation Data Base Management System.
4 XML: eXtensible Markup Language.
b) personalisation system;
c) publishing gateway;

a) The Template System
The core of the publishing phase is represented by the template system, which makes it possible to combine the content items, kept neutral so far in the central file, within the presentation that is most suitable for publication.
Unlike the word-processor document models, the templates of a CMS are programs which employ a scripting language of their own to define the logic of the publication composition.
Templates include:
• static elements: texts, images and scripts published without further processing;
• program blocks capable of retrieving content items and of displaying them in the most appropriate manner as part of the publication.
Templates differ from each other, as well as for the way in which they assemble content items, chiefly in connection with the publication channel type they are designed for. In a CMS, the management of publications on different channels also depends on the template system.

b) The Personalisation System
Prior to publication, the content is subject to the personalisation system, thanks to which the content and the way in which it is presented are defined, depending on the profile of those who will make use of it.
This obviously involves the need for the system to be able to recognise potential users. In view of this, the personalisation process can only be implemented for certain communication channels. Web publications certainly enable maximum flexibility in terms of personalisation.

c) Publishing Gateway
The Publishing Gateway is a final point through which content designed for publishing is to transit.
There are basically two types of publications:
• publications for the web, such as Internet, Intranet and Extranet websites that a CMS generates.
• Other publications. The same process employed to create web pages can also be used for other types of publications, such as electronic, press and syndication publications.

• From Web server log files. These are files in which all the requests made by users’ browsers are made.
• From application package forms. These are the forms customised to link the consultation data to the content items managed in the CMS.
Thanks to the analysis system, in particular, it is possible to increase the meta-data available for each published content item with information such as:
• Number of times in which the content has been viewed.
• Number of people who have signalled the content to other users.
• Any comments on the content itself.
The data gathered and analysed by this process are subsequently re-entered in the central file of the CMS through the Feedback Gateway. As they arrive in an already preset format and structure, the data on consultation are directly acquired without having to go through the conversion and aggregation processes.
The analysis data are fundamental for the decision-making processes on which the strategies used in the choice of content items to be published are based. By analysing the users’ fruition, you are able to evaluate, for instance, what type of content items are viewed most often and which less and based on this decide to increase the production of the most popular content items. Within the scenario of a large-sized website which decides to become multilingual, it may be beneficial to record which content items are most often viewed by foreign users and therefore only translate those, rather than having all the content items translated, thus possibly incurring significant costs.

11 Conclusion
The goal of the paper is to introduce agent based technologies and to show that they could be used to model a Content Management System, which is responsible for gathering, managing and publishing blocks of information which are regarded as content. Different agent based paradigms are reviewed and a CMS is described in detail, showing it’s a complex system and hence can be modelled using the agent based approach. In future works we’ll use UML formalisms to project an object oriented model using agents.

REFERENCES
Bensaid N, & Mathieu P. (1997), A hybrid architecture for agents
Bonabeau E., (2002). Agent-based modeling: Methods and techniques for simulating human systems, PNAS 99 Suppl. 3 (pp. 7280-7287)
AGENT BASED MODELS
A BASED WORDS AND AUTOMATA MODEL FOR INFINITELY REPEATED GAMES

M. Arfi
Université du Havre
BP 540, 76058 Le Havre, France

B. Ould M Lemine and C. Selmi
Université de Rouen
BP 12, 76801 Saint-Etienne du Rouvray, France

ABSTRACT

In this paper, we deal with modelling infinitely repeated games by using infinite words. These games are supposed noncooperative and with a perfect knowledge of the previous moves. In this context, we give a general definition of a Nash equilibrium, that we illustrate with a famous example.

Index Terms— Modelling, infinite words, formal languages, game theory, strategy, Nash equilibrium.

1. INTRODUCTION

Game theory [9] is usually defined as a mathematical tool used to analyze strategical interaction, the game, between individuals which are called players. The games studied in this paper are supposed simultaneous, noncooperative, infinitely repeated and with a perfect knowledge of the previous moves. We will elucidate these ideas through a famous example.

In game theory, the distinction between the cooperative and noncooperative game is crucial. The Prisoner’s Dilemma [4] is an interesting example to explain these notions. It is a game involving two players where each one has two possible actions: cooperate (c) or defect (d). The game consists of simultaneous actions of both players (called moves). It can be represented using the matrix:

<table>
<thead>
<tr>
<th></th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>(4,4)</td>
<td>(0,5)</td>
</tr>
<tr>
<td>d</td>
<td>(5,0)</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>

where each entry $e_{ij}$ is an ordered pair of real numbers. The two players are referred to as the row player and the column player respectively. The actions of the first player are identified with the rows of the matrix and those of the second one with the columns. If the row player chooses action $i$ and the second action $j$, the components of the ordered pair $e_{ij}$ are the payoff received by the first and the second player respectively. It is clear that if they could play cooperatively and make a binding agreement, they would both play c. If the game is noncooperative, the best action for each player is d.

Suppose now that we consider now infinite repetitions of a noncooperative base game. This game is just as noncooperative as the base one, but it allows a certain form of interaction. Suppose that each player has a perfect knowledge of the previous moves of all the others. Then his strategy may depend on these previous moves and he may coordinate it with that of his opponents. For instance, if the base game is the Prisoner’s Dilemma, grim-trigger is the strategy of cooperating in the first move and until your adversary defects, then of always defecting after the first defection of your opponent. Tit-for-tat is the strategy of playing at each step the action played by your adversary at the previous one; the initial move is free.
In this paper, we make use of infinite words to analyze the kind of games we want to model. A match of such a game is represented as an infinite word on the alphabet $A$ of moves. In this context, a strategy for player $i$ can be viewed as a relation from the set of finite words on $A$ to that of the actions of this player. The whole strategy of the game is defined as the vector composed by using the strategies of all players. We can associate to each strategy vector a language $L$ of infinite words on $A$, defined as the set of all matches that the players may make if everyone follows the strategy he decided to apply.

Nash equilibrium is one of the most important notions in games theory. The whole strategy of the game is defined as the vector composed by using the strategies of all players. Intuitively, a strategy vector is a Nash equilibrium if one player’s departure from it while the others remain faithful to it results in punishment. The idea is that once the players start playing according to such a strategy vector, then they all have a good reason to stay with it.

More precisely, our study will be organized as follows. Sections 1 contains some basic notions on game theory. In Section 2, we introduce the notion of strategy and give the definition of language generated by a strategy. Section 3 is devoted to the formal definition of a Nash equilibrium with some examples.

2. MATHEMATICAL MODEL FOR GAMES

Non cooperative games in which moves consist of simultaneous actions of $n$ players, can be represented by a collection of $n$ utility functions. The values of these functions define the expected amount paid to the players. A game is a tuple $G = (P, A, \pi)$ where
- $P = \{1, \ldots, n\}$, $n \in \mathbb{N}$, is the set of players.
- $A_i$ is the set of the actions for player $i$.
- $A = A_1 \times \ldots \times A_n$ is the alphabet of the moves.
- $\pi_i : A \rightarrow \mathbb{R}$ is the utility function for player $i$.

- $\pi = (\pi_1, \ldots, \pi_n) : A \rightarrow \mathbb{R}^n$ is the utility vector.

We consider in this paper the $\delta$-discounted infinitely repeated game of $G$, which we note by $G^\omega$. In such a game, we model a match $h$ as an infinite sequence of moves which can be represented by an infinite word on the alphabet of the moves $A : h = h_0 h_1 \cdots h_i \cdots \in A^\omega$. We denote by $h_{i,j}$ the $j^{th}$ component of move $h_i \in A$.

The utility with discounting factor $\delta \in (0, 1)$ of a match $h$ for player $i$ is defined as:

$$\pi_i^\delta(h) = (1 - \delta) \sum_{k=0}^{\infty} \pi_i^k(h_k) \delta^k.$$  

Example 2.1 As concerns the Prisoner’s Dilemma, we have $P = \{1, 2\}$, $A_1 = A_2 = \{c, d\}$, $A = \{c, d\} \times \{c, d\}$ and the utility function is defined by the matrix given in the Introduction. The infinite word $h = (c, c)^\omega$ is an example of a match in which the two players cooperate infinitely. The value of the utility function with discounting factor $\delta \in (0, 1)$ of $h$ for player $i$ is:

$$\pi_i^\delta(h) = (1 - \delta) \sum_{k=0}^{\infty} \pi_i((c, c)) \delta^k = 4(1 - \delta) \sum_{k=0}^{\infty} \delta^k = 4.$$

3. STRATEGIES AND LANGUAGES

A nondeterministic strategy, called also quasi-strategy, $\sigma_i$ is a relation from $A^*$ into $A_i$, that describes the behaviour of player $i$ during the game. A strategy vector on $A$ is the relation

$$\sigma = (\sigma_1, \ldots, \sigma_n) : A^* \rightarrow A$$

defined by:

$$\forall w \in A^*, \forall a_i \in A_i, 1 \leq i \leq n, \forall w \in A^*, a_i \in A_i, 1 \leq i \leq n.$$

Let $\Sigma$ be the set of all strategy vectors on $A$. We consider the map $\gamma : \Sigma \rightarrow \mathcal{P}(A^\omega)$, where $\mathcal{P}(A^\omega)$ denotes the set of all languages in $A^\omega$, which associates to each strategy $\sigma \in \Sigma$, the language of infinite words $\gamma(\sigma)$ given by:

$$\gamma(\sigma) = \{h \in A^\omega | h_0 \in \sigma(\epsilon) \text{ and } h_{i+1} \in \sigma(h_i) \forall i \geq 0\}.$$
\[ \cdots h_t, \forall t \geq 0 \].
The language \( \gamma(\sigma) \) represents the set of all matches that the players can play according to \( \sigma \).

**Example 3.1** We give a strategy for the Prisoner’s Dilemma game.

\[
\sigma(w) = \begin{cases} 
\{(c,c), (c,d)\} & \text{if } w \in (c,c)^* \\
\{(d,c), (d,d)\} & \text{if } w \in (c,c)^*(c,d) \\
\{(d,c) + (d,d)\} & \text{otherwise} 
\end{cases}
\]

It is usually called the "grim-trigger” strategy. The language \( L = \gamma(\sigma) \) is described by the \( \omega \)-rational expression

\[(c,c)^\omega + (c,c)^*(c,d)((d,c) + (d,d))^\omega.\]

**Example 3.2** Consider the following strategy \( \sigma \) on the alphabet \( A = \{a, b\} \):

\[
\sigma(w) = \begin{cases} 
\{a, b\} & \text{if } |w|_a < |w|_b \\
b & \text{otherwise} 
\end{cases}
\]

The language \( L = \gamma(\sigma) \) associated is

\[\{h \in A^\omega \mid Pref(h) \in \{w \in A^* \mid |w|_a \leq |w|_b \}\}.\]

We note that this language is not \( \omega \)-rational, in the sense of language theory.

In the following examples, we deal with strategies in which the players need only a finite memory to store the past moves. Such strategies can be represented by finite Büchi automata [7]. The languages associated to these strategies are \( \omega \)-rationals. We suppose that all states of the automata considered in this paper are final.

**Example 3.3** Grim-trigger strategy for player 1

The grim-trigger strategy in Example 3.1 is given by the following automaton. Notice that this automaton describes a strategy function for the first player. Indeed, all the arrows starting from each state are labelled with the same first component.

**Example 3.4** Tit-for-tat strategy for player 1

The strategy described by this automaton is in fact composed of two elementary strategy functions for the first player, which depend on the initial state chosen at the beginning of the match. We denote by \( L_c \) (resp. \( L_d \)) the set of matches played if the first player chooses action \( c \) (resp. \( d \)) to start. The language recognized by this automaton is \( L = L_c \cup L_d \), where:

\[L_c = ((c,c) + (c,d)^* (d,c))(d,d)^\omega + ((c,c) + (c,d)^* (d,c))^\omega,\]

\[L_d = ((d,d) + (d,c)^* (c,d))(c,c)^\omega + ((d,d) + (d,c)^* (c,d))^\omega.\]

**Example 3.5** Weak grim-trigger strategy for player 1

Here we have a deterministic automaton, the strategy of which is a pure relation. The behaviour of the first player becomes unpredictable after the defection of player 2. The set of matches recognized by this automaton is the language:

\[L = (c,c)^\omega + (c,c)^*(c,d)A^\omega.\]

4. NASH EQUILIBRIUM

Intuitively, a strategy vector is a Nash equilibrium if no player has any interest in leaving his strategy, while his opponents remain faithful to theirs. Let us first introduce some basic notions, in order to give a formal definition of a Nash equilibrium.
Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in A$. We call $i$-variation of $\alpha$ every $\beta \in A$ such that $\alpha_i \neq \beta_i$ and $\alpha_j = \beta_j, \forall j \neq i$.

Let $X$ be a language of $A^\omega$. We call $i$-variation of a match $h = h_0h_1 \ldots h_t \ldots$ in $X$ every match $\overline{h} \in X$ for which there exists $t \geq 0$, an $i$-variation of $h_t$ and a word $w \in A^\omega$ such that $\overline{h} = h_0 \ldots h_{t-1}ow \in X$.

A good match for player $i$ in $X$ is a match $h \in X$ verifying $\pi_i(h) \geq \pi_i(\overline{h})$ for every $i$-variation $\overline{h}$ of $h$. Denote by $GM_i(X)$ the set of all good matches for player $i$ in $X$.

**Example 4.1** Consider the language $L = (c, c)^\omega + (d, d)^\omega$. It is obvious that the words $(c, c)^\omega$ and $(d, d)^\omega$ do not admit any $i$-variation in $L$. So we have $GM_i(L) = L$, $\forall i = 1, 2$.

**Example 4.2** Let $L$ be the language of infinite words recognized by the following automaton.

![Prisoner's Dilemma Automaton](image)

This language involves the Prisoner’s Dilemma strategy in which the first player defects as far as his adversary defects and cooperates infinitely as soon as his opponent cooperates. It is clear that $L = (d, d)^\omega + (d, d)^* (c, c) + (c, d))^\omega$. We claim that $h = (d, d)(c, c)^\omega \in GM_2(L)$ if $\delta > 1/5$, otherwise $(d, d)^\omega \in GM_2(L)$. Indeed, let $\overline{h}$ be a $2$-variation of $h$. Then $\overline{h} \in (d, d)^\omega + (d, d)^* (c, c) + (c, d))^\omega$. But, at the sight of the payment matrix given in the Introduction, we notice it pays more payful for the second player always to choose $d$ instead of $c$ after his first cooperation. Thus, we will only examine the 2-variations belonging to $(d, d)^\omega + (d, d)^* (c, c) + (c, d))^\omega$. We obtain successively for $n \in \mathbb{N}$:

$$\pi_2^\delta((d, d)^n(c, c)(c, d))^\omega = (1 - \delta)(\sum_{k=0}^{n} \delta^k + \sum_{k=n+2}^{\infty} 5\delta^k) = (1 - \delta)[\sum_{k=0}^{n} \delta^k + 5(\sum_{k=0}^{\infty} \delta^k - \sum_{k=0}^{n+1} \delta^k)] = 1 - \delta^{n+1} + 5\delta^{n+2} = 1 + \delta^{n+1}(5\delta - 1)$$. 

The case of $\overline{h} = (d, d)^\omega$ can be dropped when $\delta > 1/5$, because we have $1 + \delta^{n+1}(5\delta - 1) > 1 = \pi_2^\delta((d, d)^\omega)$. Furthermore, one can easily verify that the maximum of the function $n \rightarrow 1 + \delta^{n+1}(5\delta - 1)$ is reached for $n = 0$. Hence, the word $(d, c)(d, c)^\omega$ belongs to $GM_2(L)$.

The notion of Nash equilibrium also requires the introduction of some basic strategy vectors. Let $\sigma = (\sigma_1, \ldots, \sigma_n)$ be a strategy vector and let $X = \gamma(\sigma)$ be the associated language. We denote by $\omega_i : A^* \rightarrow A_i$ the unpredictable strategy for player $i$, given by $\omega_i(w) = A_i, \forall w \in A^*$.

We define for all $1 \leq i \leq n$, the following strategy vectors:

$$\begin{align*}
\mu^{(i)} &= (\omega_1, \ldots, \omega_{i-1}, \sigma_i, \omega_{i+1}, \ldots, \omega_n) \\
\nu^{(i)} &= (\sigma_1, \ldots, \sigma_{i-1}, \omega_i, \sigma_{i+1}, \ldots, \sigma_n).
\end{align*}$$

We denote by $X_i = \gamma(\mu^{(i)})$ and by $Y_i = \gamma(\nu^{(i)})$.

**Proposition 4.3** We have:

- $X = \cap_{1 \leq i \leq n} X_i$;
- $Y_i = \cap_{j \neq i} X_j \forall 1 \leq i \leq n$.

**Proof.** For the first part, we obtain the succession of equations:

$$X = \{h \in A^\omega | h_0 \in \sigma(\epsilon) \text{ and } h_{t+1} \in \sigma(h_0 \ldots h_t), \forall t \geq 0\} = \{h \in A^\omega | h_{0,i} \in \sigma(\epsilon) \text{ and } h_{t+1,i} \in \sigma(h_0 \ldots h_t), \forall 1 \leq i \leq n, \forall t \geq 0\} = \cap_{1 \leq i \leq n} \{h \in A^\omega | h_{0,i} \in \sigma(\epsilon) \text{ and } h_{t+1,i} \in \sigma(h_0 \ldots h_t), \forall t \geq 0\} = \cap_{1 \leq i \leq n} \gamma(\mu^{(i)}) = \cap_{1 \leq i \leq n} X_i.$$ 

The second part of the proof is easier, since we immediately obtain by using the lines above:

$$Y_i = \{h \in A^\omega | h_{0,j} \in \sigma(\epsilon) \text{ and } h_{t+1,j} \in \sigma(h_0 \ldots h_t), \forall t \geq 0, \forall j \neq i\} = \cap_{j \neq i} \gamma(\nu^{(j)}) = \cap_{j \neq i} X_j.$$ 

**Definition 4.4** A strategy vector

$$\sigma = (\sigma_1, \ldots, \sigma_n)$$
is a Nash equilibrium if
\[ \bigcap_{i=1}^{n} GM_i(Y_i) \neq \emptyset. \]

In other words, a strategy vector is a Nash equilibrium if there exists a match that represents a good match for each player in the set of matches of the others [4]. In particular, in the case of two players, the general definition becomes:
\[ GM_1(X_2) \cap GM_2(X_1) \neq \emptyset. \]

**Example 4.5** We consider in the Prisoner’s Dilemma game, the vector \( \sigma = (\sigma_1, \sigma_2) \) in which both players follow the grim-trigger strategy. In this case we have:

\[ X_1 = (c,c)^{\omega} + (c,c)^{\omega}((c,c),(d,d))^{\omega}, \]
\[ X_2 = (c,c)^{\omega} + (c,c)^{\omega}(d,d)((c,d),(d,d))^{\omega}. \]

We claim that \((\sigma_1, \sigma_2)\) is a Nash equilibrium if and only if the discounting factor \( \delta \geq 1/4 \).

Indeed \((c,c)^{\omega} \in GM_1(X_2) \cap GM_2(X_1)\). Suppose \( \overline{h} = (c,c)^{k-1}(c,d)(d,d)^{\omega} \) be a match with defection of the first player at the rank \( k \geq 0 \).

We obtain after computations
\[ \pi_1^k(h) - \pi_1^k(\overline{h}) = \delta^k(4\delta - 1). \]

Then \( \pi_1^k(h) - \pi_1^k(\overline{h}) \leq 1/4 \), which proves that \( h \in GM_1(X_2) \) whenever \( \delta \geq 1/4 \). In the same way, we can show that \( h \in GM_2(X_1) \).

### 5. CONCLUSIONS AND PERSPECTIVES

This paper is essentially devoted to the study of nondeterministic strategies and Nash equilibria. Our purpose in the future consists of characterizing the family of languages generated by such strategies, called strategical languages. We also wish to study mixed strategies (with probabilities). We think that this kind of strategies involves the non deterministic ones.

### 6. REFERENCES


DEFINITION OF OPERATING PROCEDURES IN GENERIC AGENT BASED SIMULATION MODEL OF TRANSPORTATION LOGISTIC TERMINALS

Norbert Adamko
Valent Klima
Faculty of Management Science and Informatics
University of Žilina
Slovak Republic
Norbert.Adamko@fri.uniza.sk
Valent.Klima@fri.uniza.sk

KEYWORDS
Generic simulation model, agent based simulation, network charts, operating procedures, transportation terminals.

ABSTRACT
The paper presents an approach to the definition of operating procedures in simulation models of transportation terminals (e.g. marshalling yards, factories, train care centres, etc.). The presented design is based on adaptation of network charts, which provide an understandable form of operating procedure definition for model designers as well as for clients. Described approach is implemented in generic simulation tool Villon, which provides comfortable editors for definition of operating procedures, run time visualisation of customer’s attendance and also a set of post-simulation evaluations (PERT analysis) to aid the model designer.

INTRODUCTION
Transportation logistic 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. Due to the obvious advantages, generic simulation models are often utilised to support mentioned tasks. Generality of such models is, however, balanced by their increased complexity, which can lead to problems with intelligibility of created models. Operating procedures are sometimes defined/parameterised in a way that is not easily understandable (e.g. tables, dialogs at many places throughout the software) and does not allow to perceive the process of customer’s attendance as a whole (e.g. HaCon 2007).

In this text we present an approach to the definition of operating procedures that is based on network charts. Described approach is implemented in generic simulation tool Villon and provides flexible and understandable means for definition of operating procedures inside simulation models of transportation logistic terminals.

In order to ease the understanding of presented topic, some basic features and properties of the tool as well as the architecture ABASim are described first, followed by the explanation of operating procedure definition, interpretation, execution and evaluation.

VILLON SIMULATION TOOL
The simulation tool Villon (Adamko et al. 2007) allows users (professionals in the field of logistics) to create detailed simulation models of transportation terminal operation, define simulation scenarios, make experiments with the model and evaluate results of simulation runs in single integrated user-friendly environment. 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).

Model Architecture and Agent Structure
The development of Villon was based on proprietary agent based simulation architecture ABASim (Agent Based Architecture of simulation models), which provides means for design of flexible simulation models of complex service systems (initially mentioned in (Klima and Kavička 1996) and elaborated in (Kavička et al. 2007)). 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.

All communication, internal inside-agent and also external inter-agent, is realised exclusively by messages. Following types of inter-agent messages are utilised:

- **Notice** – contains some information for the addressee without expecting any answer.
- **Request** – carries specific demands, which are expected to be satisfied or supplied (e.g. resource allocation, transfer of elements, etc.).
- **Response** – delivers answer to the request, usually...
first after the request has been satisfied (the requested agent holds the message until it is able to satisfy the request and then it sends Response message to the original sender), however the response can also be negative.

Modelled system is in ABAsim architecture represented by permanent hierarchical structure of agents, which are responsible for their respective tasks; 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 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 (controls the arrival of customers), Service agent (responsible for the execution of user-defined operating procedures) and User Cooperation agent, which stands for humans interacting with the model during run-time. 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, 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**

**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**

The infrastructure (tracks, roads, etc.) model is created by direct transformation of a map (plan) – an accurate, not only simplified schematic model of infrastructure is used in the simulation. Mobile resources (e.g. personnel, engines) are modelled individually, respecting their working hours, profession and other desirable properties.

**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 (e.g. number of cars, arrival time).

**Operation**

Operating procedures are defined by the designer of the simulation model in the form of network charts and assigned to customers (each customer has one network chart that describes its operating procedure). Specialised Service agent is then responsible for their proper execution. Definition of operating procedures can be considered to be the core of the simulation system; therefore a special attention has been paid to it in Villon. The approach utilised for definition of operating procedures in Villon is described in detail in following text.

**DEFINITION OF OPERATING PROCEDURES**

The main operating procedure inside logistic terminals (similarly as in any other service system) is serving of customers entering the system. Technological procedures of serving customers differ not only between various types of transportation terminals (e.g. marshalling yards vs. locomotive depots), but also between terminals of the same type (every company has its own operating procedures). In addition, by a deeper analysis, we find out that the technological procedure differ even between customers of the same terminal (e.g. each train in a container terminal can have slightly modified handling procedure or some tasks can take longer time, etc.).

Since Villon is a generic simulation model and has to support modelling of various logistic terminal types, it is not reasonable to hard-code any pre-defined operating (service) procedures – these have to be easily defined by the designer during the creation of the model. This also means that Villon is not able to perform any task without proper “program” – the technological service procedure, defined by the model designer.

To guarantee the success of simulation study, it is vital for the designer to communicate with the clients (in most cases employees of modelled terminal). Since the clients are usually not skilled at computer simulation, it is needed to define the operating service procedures using mutually understandable form of notation.

Gantt charts are often used to describe working procedures in transportation terminals (Flodr 1990). It is a time dependent chart, so it provides an easy identification of time duration of the individual tasks as well as of the
whole procedure. However, Gantt chart does not provide clear presentation of mutual tasks dependencies, which are very important for the simulation model. In addition, at the time of the definition of service procedures for a simulation model, duration of most tasks is not known (e.g. it is not possible to know, how long it will take for a train to move from one place to another – a route could be blocked, train weight is not known, etc.).

From the simulation modelling point of view, a network chart (PERT chart, Kerzner 2003) is more suitable notation form for operating procedures. Network chart provides clear evidence of activity relationships (parallelism, dependency) and it can be constructed without knowing the actual duration of activities. Each edge of the chart is representing single activity. Vertices of the chart can be considered as synchronisation points of the operating procedure. Based on the relationship to the real tasks, activities could be divided into two groups:

- Activities representing actual task in modelled system, e.g. uncoupling of the locomotive, brake test, etc.
- Activities not reflecting any real task but bearing the information about dependencies between other activities. Such activities are called fictitious or dummy activities (see the edge [5, 6] on Fig. 2).

In Villon terminology, network charts that define operating service procedures are called Technologies. Technologies are composed of activities, which represents single tasks executed during serving of customers (e.g. loading, resource assignment, brake testing, etc.). Utilisation of network charts for definition of operating procedures in simulation models was inspired by the work of Sadlon (1994).

![Network Chart Example](image)

**Figure 2: Example of a network chart defining operating procedure of serving a customer**

**Activities definition**

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 (currently 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 can be used; profession defines the ability of the resource to execute respective tasks) 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’.

**Definition of technology**

Parameterized derived activities are composed to network charts (technologies), 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 technology. Network charts are created in a comfortable graphical editor (Fig. 3) with support for automatic validation of entered technologies (guarding required succession of some activities and appropriate resource handling).

Ready network chart is then assigned to a customer (e.g. a train or truck). Once defined technology can be reused – the same technology can be applied to different customers with the same attendance procedure.

This approach makes the modification of serving procedure of customers very flexible – user can modify parameters or resources, change mutual dependency of activities (change the network chart drawing) or even exchange whole technology at once (simply by assigning another technology from the list of defined technologies).

**EXECUTION OF OPERATING PROCEDURES**

Specialised Service agent is responsible for serving customers according to specified operating procedure (each customer that enters the system has one technology attached, which describes its attendance procedure). During simulation, the defined technologies are interpreted by Service agent. The technology depicted on figure 2 will be interpreted as follows: at first the activity Customer’s Arrival will be executed, afterwards activities Resource_A allocation and Resource_B allocation will be executed in parallel. The activity Service_B can be executed first after the activities Service_A and Resource_B transfer have been finished (the dependency is defined using dummy fictitious activity connecting vertices [5, 6]).
To ensure that all customers will be completely and correctly served Service agent can (actually have to) cooperate with other model agents responsible for execution of respective activities (e.g. activity Track assignment can be executed only by Rail agent). Because agents in ABAsim architecture communicate strictly via messages, each activity in the technology is first translated into a message. Translated technology is depicted on figure 4, the activity Resource A assignment was translated to a Request message addressed to the agent responsible for resource management with the request to assign the required resource (information about the resource Resource A is contained in the message).

Figure 4: Technology after translation by the interpreter

After translation, the execution of the technology is quite straightforward. During the simulation run, the Service agent follows the network chart (respecting the order and dependencies between activities defined in the chart structure) and simply sends respective message to the agent which carries out the activity (reflecting defined resources and parameters). Besides sending messages, Service agent is also managing responses to sent messages in order to capture finishing of activities (receiving a Response message means that the requesting activity has been finished).

EVALUATION OF OPERATING PROCEDURES

Besides animated output of modelled activities in 2D or 3D presentation (the user can see elements moving on the scene), Villon offers also the presentation of the execution status of serving procedures. User can immediately recognise current state of the attendance; colour coding makes it possible to distinguish between already executed activities, activities being executed right now and activities which will be executed in the future. By pointing the mouse on the activity, user can see all defined resources, values of all parameters as well as the starting, finishing times and duration of the activity (if the activity has been executed already).

To ease the debugging of the simulation model, user is able to set a breakpoint flag on any activity. Breakpoint can be set during definition of activities, during definition of operating procedures (technologies) and even during simulation run. In case that an activity with breakpoint flag set should be executed, simulation run is paused and user is presented with a window showing current status of the technology.

Villon also provides a set of post-run evaluations that make it possible to analyse executed operating procedures. To this purpose, a detailed protocol on simulation is
generated during the simulation run. To the available evaluations belongs PERT analysis of executed technologies (presented to the user in the form of a Gantt chart) and visualisation of critical path, which can be computed for single execution of the technology or as an aggregate evaluation for all customers sharing the same serving procedure. One of such evaluations is depicted on figure 5 (activities are presented in different colours based on the probability that the activity belongs to the critical path). This evaluation can provide user with valuable information, e.g. if a resource assignment activity is not on the critical path, it may be possible to lower the number of resources in next experiment without influencing total execution time; another example is a problematic activity sequence on a critical path, in this case the user can easily reorganize the operating procedure (using graphical editor) to make some tasks run in parallel.

Client’s understanding of the simulation model is one of the most important factors that influence the credibility of the simulation model. Proposed approach utilises naturally intelligible form of operating procedures representation (the network charts) that is easily understood by clients (who are usually not skilled in simulation or informatics) and helps to increase their confidence in the simulation model.

Having a long-term experience with practical application of one such generic simulation tool (Villon was successfully applied within many simulation studies throughout Europe and China, modelling various types of transportation systems), we are confident that proposed approach to definition of operating procedures gives the designer of the simulation model great deal of flexibility and is at the same time transparent, easily understandable and maintainable.

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


A NEW PROBABILISTIC COMBAT MODEL OF DIRECT FIRE WITH RANDOM FIRE DISTRIBUTION

Dr. B. Nguyen
M. Rempel
Centre for Operational Research and Analysis
Defence Research and Development Canada
101 Colonel By Drive
Ottawa, Canada
K1A 0K2
Email: mark.rempel@drdc-rddc.gc.ca

KEYWORDS
discrete simulation, operations research, military

ABSTRACT

In recent years, we have witnessed the growing application of agent-based simulations in military operations research, particularly in the area of combat modeling. In this paper we present a new probabilistic combat model of a direct fire exchange between two homogeneous forces, each employing a random fire tactic. This tactic emulates those typically employed in agent-based combat models. We compare our results with those of an agent-based model and show good agreement. As well, we compare our model with a Lancaster direct fire model, which assumes an uniform fire tactic, and discuss the differences between the models.

INTRODUCTION

In recent years, we have witnessed the growing application of agent-based simulations in military operations research (Iachinski, 2000; Cares, 2002; Friman and Horne, 2005). One of its main applications is in the area of combat modeling, that is a scenario of Blue forces against Red forces. The classic method of modeling combat is to employ the set of Lancaster direct fire differential equations (Taylor, 1993). While it is understood that a Lancaster direct fire model does not reflect actual combat scenarios, little attention has been paid to the differences between the Lancaster model and agent-based simulation of direct fire combat. The underlying tactics of a Lancaster direct fire model are such that weapon fire is distributed uniformly throughout the targets, while agent-based simulations tend to assume a random weapon fire distribution (Iachinski, 2000). Clearly, the two corresponding firing tactics are entirely different. As a result, their outcomes must also be different.

To understand this difference, we emulate the agent-based random firing tactics by developing a novel analytical model of many Blue forces against many Red forces where the number of shots from one side is allocated randomly to the other side. Our model is stochastic and is exact (i.e., we make no assumptions on the exchange of fire between the two sides except for the fact that the shots are randomly assigned). To the best of our knowledge, such an exact model has not been developed previously. We also believe this is an improvement to other many-to-many engagement models of direct fire where time is continuous (Taylor, 1993) or where the metrics derived assume limiting cases of the probability of single shot success (Bexfield, 1984). To build this model, we make use of combinatorics. This allows us to recognize representative allocations and differentiate those that are not representative. Doing so reduces computational time drastically.

Section 2 describes the model. The model is presented, along with a concrete example, from the viewpoint of the Blue force engaging the Red force for clarity. Section 3 discusses results from our model and presents a comparison with an agent-based model and with the Lancaster direct fire differential equations. Section 4 supplies the conclusion and discusses future work.

THE MODEL

The analytical combat model presented in this paper describes a probabilistic combat engagement of two homogeneous forces. The assumptions of the model are similar to a probabilistic direct fire model (Przemieniecki, 2000), however the distribution of weapon fire between the two forces is random rather than uniform. The assumptions are given as:

1. Both forces are homogeneous and are continually engaged in combat.
2. Each unit or individual weapon is within the maximum weapon range of all of the opposing units.
3. Each unit or individual weapon takes exactly one shot per time step.
4. The single shot probability of hitting a unit for each force is constant.
\[\rho(b, r, t + \Delta t) = \rho(b, r, t) P(0 \text{ Blue casualty in } \Delta t|(b, r)) P(0 \text{ Red casualty in } \Delta t|(b, r))
+ \rho(b + 1, r, t) P(1 \text{ Blue casualty in } \Delta t|(b + 1, r)) P(0 \text{ Red casualty in } \Delta t|(b + 1, r))
+ \rho(b, r + 1, t) P(0 \text{ Blue casualty in } \Delta t|(b, r + 1)) P(1 \text{ Red casualty in } \Delta t|(b, r + 1))
+ \cdots
+ \rho(b + 2, r + 1, t) P(2 \text{ Blue casualty in } \Delta t|(b + 2, r + 1)) P(1 \text{ Red casualty in } \Delta t|(b + 2, r + 1))
+ \cdots\]  

(2)

5. Collateral damage within the target area is negligible.

6. Each unit that is hit ceases to be engaged in combat.

7. Each unit is aware of the location and condition of all opposing units so that its fire is directed only to units that are alive (that have not been previously hit). Thus, when a valid target is hit, search begins immediately for a new target.

8. Weapon fire is randomly distributed over surviving units.

9. A single shot may not be divided across multiple targets.

Given two homogeneous forces, labeled Blue and Red with initial force strengths \(b_0\) and \(r_0\) respectively, the subsequent strengths \(b\) and \(r\) may be determined probabilistically. Thus, at time \(t + \Delta t\) from the beginning of the engagement the probability of the occurrence of any particular set of force strengths \((b, r)\) will be determined as a function of their initial force strengths, single shot hit probabilities for the Blue and Red forces, and the force strengths at time \(t\). The probability of a set of force strengths \((b, r)\) at time \(t\) is given as \(\rho(b, r, t)\), with the initial condition:

\[
\rho(b, r, 0) = \begin{cases} 
1 & \text{if } b = b_0 \text{ and } r = r_0 \\
0 & \text{otherwise}
\end{cases}
\]  

(1)

From the definition of conditional probability (Mann, 1998) equation (2) follows, where \(P(\ldots|\ldots)\) is the probability of an event described in parentheses. As shown in equation (2), a set of force strengths \((b, r)\) at time \(t + \Delta t\) may be achieved through a variety of transitions (e.g., \((b + 1, r, t) \rightarrow (b, r, t + \Delta t)\), meaning there are \(b + 1\) Blue and \(r\) Red at time \(t\) and that one Blue casualty and zero Red casualties occur in time \(\Delta t\). The probability of any given transition occurring is governed by the distribution of the Blue force shots upon the Red force and vice versa. For a given set of force strengths \((b, r)\), the possible distributions of Blue force shots upon the Red force is described by the set \(\alpha(b, r)\), defined as:

\[
\alpha(b, r) = \{\vec{\alpha} | 0 \leq \alpha_i \leq b, |\vec{\alpha}| = r, \sum_{j=1}^{r} \alpha_j = b\}
\]  

(3)

where the vector \(\vec{\alpha}\), called here a shot vector, is defined as:

\[
\vec{\alpha} = [\alpha_1 \alpha_2 \cdots \alpha_r]
\]  

(4)

where \(\alpha_i\) represents the number of shots by the Blue force \(b\) assigned to target \(i\) of the Red force. The first defining-property in equation (3) states that a target may not be assigned a number of shots greater than the number of shooters; the second defining-property states that every target must be represented in the shot vector; and the third property states that the sum of the shots must equal the number of shooters. As an example, for the set of force strengths \((3, 3)\), the set of shot vectors for the Blue force are shown in the left column of Table 1. The first entry is the vector \([0 \ 0 \ 3]\), which is read as zero shots are assigned to the first target of the Red force, zero shots are assigned to the second target, and three shots are assigned to the third target.

Within the set \(\alpha(b, r)\), there are subsets of vectors that generate identical casualty probabilities \(P(\ldots|\ldots)\). As an example, for the set of force strengths \((3, 3)\) the \(\vec{\alpha}\) shot vectors \([0 \ 0 \ 3], [0 \ 3 \ 0]\) and \([3 \ 0 \ 0]\) generate identical casualty probabilities. Each set of shot vectors that generate identical casualty probabilities may be represented by a single vector called a representative shot vector. For example, the representative shot vector for the three shot vectors stated above may be given as \([0 \ 0 \ 3]\). In general, the set of representative shot vectors for a set of force strengths \((b, r)\) is given as:

\[
\gamma(b, r) = \{\vec{\gamma} | \gamma_{i-1} \leq \gamma_i, 1 \leq i \leq \gamma_r\}
\]  

(5)

where the vector \(\vec{\gamma}\) is a representative shot vector and is given as:

\[
\vec{\gamma} = [\gamma_1 \ \gamma_2 \ \cdots \ \gamma_r]
\]  

(6)

where \(0 \leq \gamma_i \leq b, \sum_{j=1}^{\gamma_r} \gamma_j = b,\) and \(\gamma_0 = 0\). The defining-properties in equation (5) combine to state that the number of shots assigned to a target are in ascending order. As an example, the set representative shot vectors \(\gamma(3, 3)\) are shown in the second column of Table 1.

As shown in Table 1, the number of shot vectors for the set of force strengths \((3, 3)\) is ten while the number of representative shot vectors is three. In general the growth of the number of shot vectors is faster than number of representative shot vectors. The number of representative shot vectors is given by the coefficient of \(x^r\) of the polynomial \(p(x)\):
Table 1: Example calculations for initial force strengths \( b_0 = 3 \), \( r_0 = 3 \), and single shot hit probability \( h = 0.2 \) for Blue and Red force at time \( t + \Delta t = 1 \). The vectors shown are for the set of force strengths \((b, r) = (3, 3)\).

<table>
<thead>
<tr>
<th>( \alpha(3,3) )</th>
<th>( \gamma(3,3) )</th>
<th>( \lambda(3,3) )</th>
<th>( M(3,3, \bar{\lambda}) )</th>
<th>( T(3,3) )</th>
<th>( \rho(b, r, 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 3</td>
<td>[0 0 3]</td>
<td>[2 0 0 1]</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0000 0.0000 0.0003 0.0004</td>
</tr>
<tr>
<td>0 1 2</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>0.0000 0.0028 0.0229 0.0270</td>
</tr>
<tr>
<td>0 2 1</td>
<td>[0 1 2]</td>
<td>[1 1 1 0]</td>
<td>6</td>
<td></td>
<td>0.0003 0.0229 0.1887 0.2224</td>
</tr>
<tr>
<td>1 0 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0004 0.0270 0.2224 0.2621</td>
</tr>
<tr>
<td>2 0 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>[1 1 1]</td>
<td>[0 3 0 0]</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
p(x) = \prod_{n=1}^{b} \left( \sum_{i=0}^{\left\lfloor \frac{x}{n} \right\rfloor} x^i \right)_{r} (7)
\]

where \( \left\lfloor \frac{x}{n} \right\rfloor \) represents the largest integer that is less than or equal to \( \frac{x}{n} \). The derivation of this formula is given by a generating function and is described in a working paper by Nguyen and Rempel (Nguyen and Rempel, 2008).

Figure 1 shows an example of the number of shot and representative vectors as a function of force strength for the situation where the Blue and Red force strengths are equal. Clearly, computing the casualty probabilities using the representative vectors rather than the shot vectors would reduce the computational time significantly.

Each representative shot vector has a corresponding shot frequency vector that describes the number of targets that have a specified number of shots assigned to them (i.e., \( x \) targets have zero shots assigned, \( y \) targets have one shot assigned, \( z \) targets have two shots assigned, etc.). For a given set of force strengths \((b, r)\), the set of shot frequency vectors for the Blue force are given as:

\[
\lambda(b, r) = \{ \bar{\lambda}\lambda_1 = \sum_{j=1}^{r} \delta_{\lambda_{j,i-1}}, \gamma \in \gamma(b, r) \} (8)
\]

where \( \delta_{\lambda_{j,i-1}} \) is the Kronecker delta function and \( \bar{\lambda} \) is a shot frequency vector given as:

\[
\bar{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{b+1}] (9)
\]

where \( \lambda_1 \) represents the number of targets that have zero shots assigned, \( \lambda_2 \) represents the number of targets that have one shot assigned, etc. The defining-properties in equation (8) state that for each representative shot vector there exists one shot frequency vector, where each element represents the number of targets that have a specified number of shots assigned. As an example, the set of shot frequency vectors \( \lambda(3,3) \) for the Blue force are shown in Table 1. For the representative shot vector \([0 0 3]\) the shot frequency vector is \([2 0 0 1]\), which is read as two targets have zero shots assigned, zero targets have one shot assigned, zero targets have two shots assigned, and one target has three shots assigned.

As mentioned, each representative shot vector describes a set of shot vectors in the set \( \alpha(b, r) \). Consequently, knowing the number of shot vectors that each representative shot vector depicts and the number of total shot vectors, the probability of each representative shot vector occurring may be calculated. Calculating the number of shot vectors that a representative shot vector describes is akin to calculating the number of arrangements of \( r \) objects with \( \lambda_1 \) of type 1 (zero shots assigned), \( \lambda_2 \) of type 2 (one shot assigned), \( \ldots \), and \( \lambda_{b+1} \) of type \( b \) (\( b \) shots assigned), with the sum of the types equal to \( r \). The number of shot vectors that each representative shot vector represents, here called the multiplicity, is given as (Tucker, 1984):
\[ M(r, \lambda) = \left( \begin{array}{c} r \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_b+1 \\ r! \end{array} \right) \left( \begin{array}{c} r - \lambda_1 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{b+1} \\ r! \end{array} \right) \]  
\[ \cdots \left( \begin{array}{c} r - \lambda_1 - \lambda_2 - \cdots - \lambda_b \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{b+1} \\ r! \end{array} \right) \] 
\[ = \frac{3!}{20!9!1!} = 3 \]  
(10)

The multiplicity of the representative shot vectors for the set of force strengths (3, 3) are shown in Table 1. As an example, the multiplicity for the representative shot vector \([2 \ 0 \ 0 \ 1]\) is given as:

\[ M(3, [2 \ 0 \ 0 \ 1]) = \frac{3!}{20!9!1!} = 3 \]  
(11)

It may be noted that distributing Blue force shots upon Red force targets is the same process as distributing \(b\) identical objects into \(r\) different boxes. This is a well known problem in combinatorics of identical distributions of objects with repetitions (Tucker, 1984), and thus the total number of shot vectors for a set of force strengths \((b, r)\) is given as:

\[ T(b, r) = \binom{r+b-1}{r} \]  
(12)

For the set of force strengths (3, 3), the number of shot vectors is ten, as shown in Table 1.

The probability of a single shot hitting its target is \(h\), and correspondingly the probability of a single shot missing its target is \(m = 1 - h\). Given these we define the following:

\[ m_i = m^i \]  
(13)
\[ h_i = 1 - m_i \]  
(14)

where \(i\) represents a number of shots, \(m_i\) represents the probability of missing a target given \(i\) shots, and \(h_i\) represents the probability of hitting a target at least once given \(i\) shots.

Using the above definitions and the definition of a representative shot vector, we may generate the probability of 0, 1, 2, \cdots r Red force casualties for a representative shot vector recursively. Starting with the first target in a representative shot vector, the probability of zero casualties is simply \(m_{r_1}\), the probability of one casualty is \(h_{r_1}\), and the probability of more than one casualty is zero. With the assessment of the second target, the probability of zero casualties is \(m_{r_1} m_{r_2}\), the probability of one casualty is \(m_{r_1} h_{r_2} + h_{r_1} m_{r_2}\), the probability of two casualties is \(h_{r_1} h_{r_2}\), and the probability of more than two casualties is zero. The probability of any particular number of casualties \(c\) requires the full assessment of a representative shot vector and may be stated by the recurrence relation:

\[ p(c, \gamma) = p(c - 1, \gamma_{i-1}) h_{\gamma_i} + p(c, \gamma_{i-1}) m_{\gamma_i} \]  
(15)

where \(p(c, \gamma) = 0\) if \(c > i\) or \(c < 0\) and \(p(0, \gamma_0) = 1\).

Using the casualty probabilities, we may calculate the conditional probability of a number of Red force casualties occurring given a set of force strengths \((b, r)\) as follows:

\[ P(c|b, r) = \frac{1}{T(b, r)} \sum_{\gamma \in \gamma(b, r)} M(r, L(\gamma)) p(c, \gamma) \]  
(16)

where the function \(L(\gamma)\) calculates the \(\lambda\) vector for the given \(\gamma\) vector. As an example, the probability of zero Red force casualties given the set of force strengths (3, 3) with a single shot hit probability of \(h = 0.2\) is given as:

\[ P(0|(3, 3)) = \frac{1}{10} (3p(0, \gamma^{(1)}_3) + 6p(0, \gamma^{(2)}_3) + p(0, \gamma^{(3)}_3)) \]  
\[ = \frac{1}{10} (3(m_{r_1}) m_{r_2} m_{r_3} + 6(m_{r_1} r_2) m_{r_2} m_{r_3}) \]  
\[ + (m_{r_1} m_{r_2} m_{r_3} + 6(m_0 m_1 m_2)) \]  
\[ = 0.5120 \]  
(17)

where \(\gamma^{(i)}_i\) is the \(i\)th element of the \(i\)th \(\gamma\) in the second column in Table 1. Similarly computed as above, \(P(1|(3, 3)) = 0.4344, P(2|(3, 3)) = 0.0528, \) and \(P(3|(3, 3)) = 0.0008\).

Recalling equation (1), we may now use equation (16) to calculate the probability that a set of force strengths \((b, r)\) exist at time \(t + \Delta t\), which is given as:

\[ \rho(b, r, t + \Delta t) = \sum_{i=0}^{b_0} \sum_{j=0}^{r_0} p(i, j, t) P_B(i-b(|i, j)) P_B(j-r|i, j)) \]  
(18)

where \(P_B(j-r|i, j)\) is the conditional probability as shown in equation (16) of a number of Red force casualties. The number of Blue force casualties \(P_B(i-b(|i, j))\) is defined in a similar manner to equation (16) and is given as:

\[ P_B(c|(b, r)) = \frac{1}{T(r, b)} \sum_{\gamma \in \gamma(b, r)} M(b, L(\gamma)) p(c, \gamma) \]  
(19)

The index of the sums in equation (18) are limited between \(b\) and \(b_0\) and \(r\) and \(r_0\) respectively, which means that transitions from sets of force strengths such as \(2, 2 \rightarrow 3, 2\) are not allowed (i.e., force strengths do not increase over time). As an example, the values for \(\rho(b, r, 1)\) with initial conditions \(b_0 = 3, r_0 = 3, \) and single shot hit probability \(h = 0.2\) for the Blue and Red force are shown in the right column in Table 1. The entries in the matrix represent the probability of a particular set of force strengths at time one, where each row represents a value of \(b\) ranging from zero to three and each column represents a value of \(r\) ranging from zero to three (e.g., the probability of the set of force strengths (2, 3)

320
existing at time one is 0.2224). It should be noted that the summation of the $\rho$ matrix at each time step is one.

Given the matrix $\rho$ at time $t$ the expected number of Blue and Red that are engaged in combat after $t$ time steps is given as:

$$E_B(t) = \sum_{i=1}^{b_0} \sum_{j=1}^{r_0} \rho(i, j, t)$$  \hspace{1cm} (20)

$$E_R(t) = \sum_{i=1}^{r_0} \sum_{j=1}^{b_0} \rho(j, i, t)$$  \hspace{1cm} (21)

where $E_B(t)$ is the expected number of Blue and $E_R(t)$ is the expected number of Red. As an example, for the probability matrix shown in the right column of Table 1 the expected value of Blue and Red each are 2.4576.

RESULTS AND DISCUSSION

We computed the expected Blue and Red force strengths, as described in the previous section, in two scenarios. The first scenario is the example presented in the previous section, where the initial force strengths are $b_0 = 3$ and $r_0 = 3$, the single Blue shot hit probability $h^{(b)} = 0.2$, and the single Red shot hit probability $h^{(r)} = 0.2$. The expected force strengths are shown in Figure 2 (a) and labeled Direct random blue and Direct random red. It should be noted that the Blue and Red force strengths are identical, and thus the Red force is plotted on top of the Blue force.

We designed an agent-based model of a direct fire scenario with random fire using the MASON toolkit (Luke et al., 2003; Luke et al., 2004). While agent-based models are typically used to model the interactions of autonomous agents that have the ability to modify their behavior based on their experiences (Macal and North, 2006), the direct fire agent-based model presented in this work is a basic model with static agent characteristics. Using a basic model as a starting point for analysis, as performed by McIntosh and Lauren (McIntosh and Lauren, 2007), allows for investigation of the firing tactics, which is the focus of the model described in the previous section.

Each member of the Blue and Red forces were represented by individual agents. Each agent was defined to be in one of two states, either alive or dead. At the beginning of a simulation all agents were in the alive state and only transitioned to the dead state when they were hit by a shot. Agents in a dead state no longer were engaged in the battle. During each time step all agents in the alive state performed three actions in the following order: 1) determine the set of opposing agents that are alive, called situational awareness; 2) select an agent randomly from their situational awareness and fired their weapon at that agent; and 3) if shot at, evaluate if the shot hit and subsequently update their state. All agents, selected randomly, completed the first action prior to performing the second action, and likewise for the third action.

In a single agent-based simulation, any valid random distribution of fire may occur at time $t$ for a given set of force strengths $(b, r)$. Thus, due to this stochastic nature we executed 10,000 simulations and computed the mean force strength for the Blue and Red forces at each time step. The results for the scenario described in the previous section are shown in Figure 2 (a) and are labeled ABM blue and ABM red. The standard error of the force strengths are not shown since they are smaller than the size of the symbols used in the figure.

We also computed the force strengths for the scenario using the Lanchester direct fire differential model. The Lanchester
model is described by a pair of coupled, linear, first-order differential equations given by (Przemieniecki, 2000):

\[
\frac{db}{dt} = -h(b)r \\
\frac{dr}{dt} = -h(r)b
\]  

(22) (23)

where \( b(0) = b_0 \) and \( r(0) = r_0 \). These equations relate the casualty rate of one force with the strength of the opposing force, and use the assumptions listed in the previous section, with the exception of the random weapon fire assumption. The shot distribution in the Lanchester model is assumed to be uniform, and thus a single shot is infinitely divisible. The force strengths calculated using the Lanchester direct fire model are shown in Figure 2 (a) and labeled Lanchester blue and Lanchester red. It should be noted that the Blue and Red force strengths are identical, and thus the Red force is plotted on top of the Blue force.

Inspection of Figure 2 (a) shows that the expected force strengths determined by our probabilistic model agrees with the mean force strengths of the agent-based simulation. When compared with the Lanchester model, there is agreement for the first three time steps, however the force strengths calculated using the Lanchester model quickly deviate and tends toward zero while the other two models asymptotically approach a value slightly below one. The asymptotic behavior reflects the fact that the two forces initially have equal force strengths and single shot hit probabilities. Therefore, there is a 50% chance that each force will win the battle, resulting in a non-zero asymptotic force strength. The Lanchester model does not demonstrate this reality primarily because the model assumes a single shot is infinitely divisible across a set of targets. This results in a continuous distribution of force strengths, rather than the discrete distribution used by our probabilistic model and the agent-based model, and subsequently the force strengths tend to zero.

As a second example, we present a scenario with an initial Blue force strength \( b_0 = 15 \) that is greater than the initial Red force strength \( r_0 = 10 \) and the two forces have equal single shot hit probabilities \( h(b) = h(r) = 0.2 \). The expected force strengths using our model are shown in Figure 2 (b) and labeled Direct random blue and Direct random red. Similar to the previous example, we calculated the force strengths using 10,000 simulations of the agent-based model, with the mean force strengths labeled as ABM blue and ABM red, and using the Lanchester model, which are labeled as Lanchester blue and Lanchester red.

The results shown in Figure 2 (b) show good agreement between our model and the agent-based model. The Lanchester fire model shows that the force strength of the Red force reaches zero at the fourth time step, which signifies that the battle ceases. This result clearly differs from the other two models for the primary reason that a uniform shot distribution produces a greater casualty rate than a random shot distribution, thus causing the Blue force to win the battle quicker. This demonstrates the difference between the two models’ firing doctrine and clearly shows that the logic in an agent-based combat model, which typically use random firing tactics, may not be described by a Lanchester model.

**CONCLUSION**

In this paper we presented a new probabilistic combat model of a direct fire exchange with random shot distributions between two homogeneous forces. We compared our model with an agent-based combat model of a direct fire engagement, assuming a random shot distribution, and showed good agreement between the two models. We also compared these results with those of a Lanchester direct fire differential equation model. Through two examples, we highlighted two key assumptions of the Lanchester model: weapon fire is uniformly distributed and a single shot is infinitely divisible. We demonstrated that a Lanchester direct fire model does not accurately reflect the logic of agent-based models of direct fire engagements, which typically assume that weapon fire is random.

The model presented may be extended to investigate more complex combat scenarios. Two examples are: 1) Blue and Red forces employing different firing tactics, for example Blue using a salvo tactic, that is as near as uniform fire as possible, and the red force using a random tactic; and 2) both forces having limited weapon ranges such that the entire opposing force is not always within range of every shooter. The first example requires a relaxation of the assumptions and a change in the definition of a shot vector, such that the shot distribution is as even as possible. In the second example the model requires greater modification, such as the definition of the shot and representative vectors, however the framework for the analysis remains. Thus, given the scenarios presented in this paper and the examples of further model development, the probabilistic combat model presented in this work represents a significant step forward in the analysis of agent-based combat models.

**References**


**AUTHOR BIOGRAPHY**

**DR. BAO NGUYEN** is a senior Defence Scientist at the Centre for Operational Research and Analysis at Defence Research and Development Canada. He holds a PhD in theoretical particle physics from McGill University, BSc in physics, and a BSc in mathematics from the University of Ottawa/California Institute of Technology. He has received many awards including a Gold Medal from the University of Ottawa, a Deputy Minister Commendation for his work on NORAD long range beacons and recently a National Practice Prize for the most outstanding application (on autonomous underwater vehicles) of Operations Research from the Canadian Operational Research Society. Dr. Nguyen has worked in several environments including NORAD & former US Space Command and US Northern Command, NATO Undersea Research Centre, and Defence Research and Development Atlantic.

**MARK REMPEL** holds a BSc in engineering physics (1999), BSc in computer science (1999), and a MSc in experimental condensed matter physics (2003) from the University of Saskatchewan in Saskatoon, Saskatchewan. He is currently a Defence Scientist at the Centre for Operational Research and Analysis at Defence Research and Development Canada in Ottawa, Canada. He received several scholarships from the University of Saskatchewan during his graduate and undergraduate studies, and was nominated for a best paper award in the Centre for Operational Research and Analysis in 2006. Prior to obtaining his MSc, he was a control systems engineer at Nortel Networks in Ottawa, Canada.
PARALLEL PROCESSING
Hardware Acceleration for Computational Intelligence - THSOM Neural Network on x86 hardware

Rudolf Marek, Miroslav Skrbek
Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague
email: marekr2@fel.cvut.cz, skrbek@fel.cvut.cz

KEYWORDS
Neural Networks, Self-Organizing Maps, Computational Intelligence, Hardware Acceleration

ABSTRACT
This paper presents an accelerated simulation of a Temporal Hebbian Self-organizing map (THSOM) neural network on x86 based platform. The THSOM computational complexity grows rapidly for larger networks requiring a lot of time to simulate the learning/recall phase of the network. To reduce the time, our implementation utilizes all parallel features of modern x86 hardware – the data parallelism using the SIMD SSE instruction set and instruction parallelism utilizing multiple cores. The overall design of our THSOM implementation is modular, allowing us to re-implement specific parts of computations with different optimizations or parallel approaches yet still maintain good comparability between different optimizing combinations. We present the results of our measurements and influence of data parallel and instruction parallel processing compared to differently optimized versions. We also present an efficient method for frequent barrier synchronization of different threads.

Introduction
We present a C implementation of a THSOM (Koutnik and Snorek 2008) neural network. The THSOM neural network is an enhanced variant of SOM (Kohonen et al. 2001) neural network with additional temporal layer, which allows us to clusterize the spatiotemporal data sets.

There were numerous software or hardware parallel implementations of the SOM neural networks. The multi-thread parallel implementation of SOM was done in 2000 (Rauber et al. 2000), on SGI Power Challenge and SGI Origin 2000 and dual Celeron platforms. Hardware implementations were designed using the re-programmable RAPTOR2000 platform (Franzmeier et al. 2004) or and as IP cores (Porrmann et al. 2003) or (Hendry et al. 2003).

Nowadays, the mainstream x86 architecture offers enhanced floating point vector computations supported through the SSE instruction set. Multiple processors (cores) in the single chip are becoming standard. They are delivering great amount of parallel processing power to common desktop/server systems. In this paper, we present a way how to utilize the parallel capabilities of modern x86 CPUs and how to solve the intensive inter-thread synchronization during the parallel computation.

Introduction to THSOM Neural Network
The Temporal Hebbian Self-organizing Map (THSOM) was introduced in (Koutnik and Snorek 2008). Unlike classic SOM (Kohonen et al. 2001) based networks, they contain spatial and temporal maps, extending the clustering to space and time. The THSOM architecture consists of one layer (usually connected as the grid) of hybrid neurons as depicted on Figure 1. They are fully connected to the input vector of dimension $d$, connections make up the spatial map. The neurons are connected to each other in the grid using recurrent temporal synapses (temporal map). Hybrid neurons contain two types of similarity measures, Euclidean metric for measuring similarities in input spatial space and scalar product for measuring similarities in temporal space. The activation (output) of a neuron is defined as follows:

$$y_i^{t+1} = \sqrt{D} - \sqrt{\sum_{j=1}^{d}(x_j^t - w_{ij}^t)^2 + \sum_{k=1}^{n}(y_k^t m_{ik}^t)} \quad (1)$$

where $y_i^{t+1}$ is activation (output) of $i$-th neuron in time $t + 1$ (next time step), $\sqrt{D}$ is a constant square root of
input vector dimension $D$, $x^j_t$ is input of $j$-th vector in time $t$, $w^{ij}_t$ is the spatial weight for $j$-th input in time $t$, $y^i_k$ is output of $k$-th neuron in time $t$, $m^{ik}_t$ is temporal weights for $k$-th neuron in time $t$ (from neuron $k$ to neuron $i$), $n$ is number of of neurons in network, and $d$ is a dimension of input space (vector).

The spatial weights are updated using classical SOM (Kohonen’s Self-organizing Map) rules (Kohonen et al. 2001). The weights are updated in order to move the neuron in the space closer to the input vector. Temporal weights use modified Hebbian learning rule according to the following formula (Koutnik 2008) (after new Best Matching Unit (BMU) b is computed):

$$m^{ik}_{t+1} = \begin{cases} \min(\max(m^{ik}_t + \alpha(1 - m^{ik}_t + \beta),K_l),K_h), & \text{for } k = \arg\max_i(y^i_k) \\ \min(\max(m^{ik}_t - \alpha(m^{ik}_t + \beta),K_l),K_h), & \text{otherwise} \end{cases}$$

(2)

where $y^i_k$ is an output of $k$-th neuron of previous time step, $\beta$ and $\alpha$ control temporal synapses learning rate, $\alpha$ can start on 1.0 and is decreased in time to slow down oscillation around desired temporal weight value, $K_l$ is a low boundary of temporal weights, usually ($K_l = 0$), $K_h$ is a high boundary of temporal weights, usually ($K_h = 1$).

Optimizing Reference C Implementation

We have chosen an artificial data set for our benchmarks. The dataset contains a simulated GPS data from a car ride across the city, sampled with constant rate. There is 2216 data points in normalized 2D coordinates. The learning phase is fixed at 1000 Epochs to ease the comparability of the different optimizations.

Table 1: The summary of top profiled functions, findBMU_normal as most complex and cutgauss2D as most called.

<table>
<thead>
<tr>
<th>% time</th>
<th>calls</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>92.50</td>
<td>2216001</td>
<td>findBMU_normal</td>
</tr>
<tr>
<td>3.03</td>
<td>2216000</td>
<td>updateSOM</td>
</tr>
<tr>
<td>2.72</td>
<td>141824000</td>
<td>cutgauss2D</td>
</tr>
<tr>
<td>1.15</td>
<td>15750045</td>
<td>gamma2D</td>
</tr>
<tr>
<td>0.18</td>
<td>231860</td>
<td>updateTSOM</td>
</tr>
</tbody>
</table>

We profiled our reference C implementation with the GNU gprof tool. We used the GCC 4.1 compiler (32-bit) together with standard optimizing options (-02 -msse2). The Table 1 summarizes the results from the THSOM network populated with 64 neurons. The results of this benchmark were used for further optimizations as described later.

The profiled data indicate expected results, that the most intensive function is the findBMU_normal, where the best matching unit computation takes place. The third most computationally intensive function is cutgauss2D, which computes the Gaussian neighborhood. Both functions are good candidates for optimization.

The findBMU_normal function can be made parallel on the instruction level as well as the data level. The parallelism on the instruction level can be achieved with multiple threads calculating the best matching unit on some subset of neurons.

The data parallel operation (SIMD) can be achieved using the SSE instruction set on multiple neurons grouped and processed in one thread. The code fragment below depicts the grouping:

```c
struct neuron4 {
    float m[NEURONS][4]; /* temporal map */
    float w[VDIM][4];    /* spatial map */
}
struct neuron4 neurons4[NEURONS / 4];
```

The grouping by four is architecture specific, because the SIMD SSE instruction can operate on four floats concurrently. The spatial layout of arrays $m$ and $w$ group each neuron weight elements to the consecutive memory space, allowing us to load the vector register without a stride.
Data parallelism using the vector SSE instructions

The compiler vector extensions were used to avoid the need for direct assembly coding of SSE instructions. The GCC compiler vector extensions operate on special data type which holds the elements of the vector.

```c
#include <xmmintrin.h>
typedef float v4sf __attribute__((vector_size(16)));
```

The declared v4sf type holds the four float elements together. The modification of the findBMU_normal function was straightforward, the inner-most loop cycling through packed four neurons was removed and substituted with vector operations. Namely the load store operations of vector cannot be coded using standard C operators. The intrinsic functions of compiler must be used (\_\_mm\_load\_ps resp. \_\_mm\_store\_ps). This optimization is labeled as \_\_DsSE in further text.

The instruction based parallelism using threads

As we already outlined, the best matching unit computation might be spread to different CPU cores using threads. Each thread computes the BMU of given subset of neurons. After all threads find its BMU, a “winning” neuron is computed from the partial BMUs obtained from the threads.

There must be a synchronization barrier for the start and stop of the parallel BMU computation. Use of a traditional barrier scheme involving sleep/wake semantics causes substantial overhead, rendering the parallel BMU computation useless.

We propose our own barrier synchronization primitive implemented with help of busy waiting and atomic arithmetic, which has no performance issues induced by sleep/wake semantics.

The thread enters the barrier by calling the barrier function with its own ID and pointer to shared data structure holding the count of threads currently spinning in the barrier.

Once the thread enters into the barrier function, the barrier is atomically increased. Depending on thread ID, the thread either spins until the barrier is zero again or if ID is 0, then it monitors the barrier. If all threads are spinning inside the barrier it will write zero into barrier – unlocking the barrier for other threads.

To avoid exclusive cache line state for any other unrelated data other than the barrier, the barrier data structure shall be padded to exactly one cache line.

The atomic barrier increase operation was implemented with the help of a multiprocess libatomic-ops library.

The Lookup Table Optimization

The cutGauss2D function results can be cached in the lookup table to speed-up the computation. The result of the cutGauss2D is stored in the 2D array for each two neurons. If the size of the neighbourhood changes then the whole lookup table is invalidated. This optimization is labeled as \_\_DnyCachein further text.

The Loop Unrolling Optimization

There are many loops in the code iterating through all neurons, especially in the findBMU function. This is an ideal opportunity for the loop unrolling technique, which will enable better parallel use of multiple function units inside the CPU. The loop unrolling can be done by hand, or by compiler. We tried both techniques.

For the manual loop-unrolling, the most critical part of the code must be identified. This time, we used the Oprofile profiler, which collects the profile data with the help of processor performance counters. The resulting profiler data consists of annotated assembly code with the number of samples and total percentage for whole program.

It turned out, that the best candidate for loop unrolling is the temporal map computation loop, which iterates on all temporal weight of each neuron (see the second sum in (1)).

We unrolled the loops four and eight times. Depending on neural network size, the additional speedup was up to ten percent. The main factor that influences the result is compiler generated code. Upon inspection, the compiler correctly scheduled the load and arithmetic instruction with proper distances but failed to generate proper optimal code for allocation of registers inducing loads to same register right after an arithmetic operation, which stall the CPU pipeline.

The automatic loop unrolling is turned on with compiler option \_\_funroll-loops . Achieved results are presented in a further section.

The Memory Prefetching Optimization

Modern x86 offers a prefetching instruction, which may load the cache line in advance minimizing the cache misses. We inserted a GCC intrinsic function \_\_mm\_prefetch into our unrolled loop cycle, but the speedup was minimal or a bit negative depending on the neural network size.

The GCC compiler is capable of inserting the prefetch instruction into code automatically. The compiler option is \_\_fprefetch-loop-arrays . The GCC documentations suggest that benefit of this option is highly dependent to structure of loops. The obtained results are presented in the next section.
Single-thread Measurements

In this section, we present measurements of a single-thread, optimized version of our THSOM implementation. The platform used was equipped with Intel quad core Core2 CPU running at 2.4GHz. The installed operating system was Debian Lenny, with a 2.6.24-686 kernel. The system compiler was GCC 4.2.3 (Debian 4.2.3-5).

We measured the THSOM performance on wide range of neural network sizes – from very small networks with only four neurons up to huge networks with 4096 neurons.

We started with unoptimized version and we compared a speed-up to compiler optimized version on level 2 and SSE math (-O2 -msse2), which is considered a good starting point for further optimizations. The average speedup was 3.7 times, which is quite significant and it can be gained with just compiler switches.

Table 2: The walltime results in seconds for various network sizes with bare -O2 -msse2 optimization

<table>
<thead>
<tr>
<th>Neurons</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.89</td>
</tr>
<tr>
<td>16</td>
<td>3.98</td>
</tr>
<tr>
<td>64</td>
<td>23.04</td>
</tr>
<tr>
<td>256</td>
<td>236.29</td>
</tr>
<tr>
<td>1024</td>
<td>3502.57</td>
</tr>
<tr>
<td>4096</td>
<td>66466.37</td>
</tr>
</tbody>
</table>

Our performance measurements are organized as follows: The Table 2 summarizes the achieved wall clock times of our benchmark task with basic compiler optimization (-O2 -msse2). All other measurements are presented as speedup relatively to the values obtained with this measurement. The measurement with compiler optimization level 3 is denoted as -O3. The SSE vector optimization is denoted as -DSSE and the loop unrolling optimization and data prefetching optimization are denoted as -funroll-loops and -fprefetch-loop-arrays. The -msse2 flag is used in all measurements – we want the compiler to generate the SSE instructions for floating point operations instead of legacy x87 FPU operations.

The Figure 2 depicts the speedups of different proposed optimizations. The -O3 optimization is comparable to reference (-O2) optimization. The average speedup is just 1.04x. If we optimize further with the help of data parallel SIMD optimization (-DSSE) we gain a significant average speedup of 2.0x. If we turn-on the loop unrolling (-funroll-loops), we gain an average speedup of 2.5x, greatly enhancing performance for networks up to 128 neurons. However the performance scales well up to 4096 and the speedup remains nearly constant. If one changes the loop-unrolling optimization to the memory loop prefetching optimization -fprefetch-loop-arrays the speedup will match the speedup from previous optimization with the exception of networks sized with 256 neurons, where the prefetching works ideally and removes the cache misses. If we combine the last two optimizations together we receive nice speedup around 3x for the range from 16 up to 1024 neurons.

Multi-thread Measurements

The multicore version was measured on same the platform. Table 3 and Figure 3 summarizes the achieved results. The speedup is relative to the fastest single thread version with same optimization flags turned on in the multithread version (-O3 -DSSE -DyCACHE -funroll-loops -fprefetch-loop-arrays).

Table 3: The speedups with the two and four thread version

<table>
<thead>
<tr>
<th>Neurons</th>
<th>2 threads</th>
<th>4 threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.39</td>
<td>0.26</td>
</tr>
<tr>
<td>64</td>
<td>0.75</td>
<td>0.71</td>
</tr>
<tr>
<td>256</td>
<td>1.34</td>
<td>1.73</td>
</tr>
<tr>
<td>1024</td>
<td>2.43</td>
<td>4.19</td>
</tr>
</tbody>
</table>

The multithread version gets faster for networks whose size is bigger then 64 neurons. For smaller networks, findBMU computation takes less time than the cost of mutual synchronization of the threads. The speedup scales well up to 1024 neurons, which fits the CPU at most gaining even a little better then 4 times speedup. This may be explained by cache effects.

The networks bigger than 1024 neurons scale much worse due to increased pressure on memory subsystem which brings down the performance. The table 4 depicts
the memory sizes for different networks per neuron and per network.
As we stated, we used a CPU with 4MB of cache total, which explains why the network with 1024 neurons
get best results. All data can fit in the cache, only a minimum of data is missed from the L2 cache.

Conclusion

We conclude that compiler based optimizations together with additional vector based optimization bring significant improvements to the speed of computations. The speedup between no optimization and GCC standard compiler optimization level 2 (with SSE math) is 3.7 times in average.
If we combine our vector optimizations together with compiler generated loop unrolling and memory prefetching we gain an average additional speedup of 2.91 for single threaded computation.
The multithread version of our THSOM implementation scales best for network with 1024 neurons. In this ideal case, when all data fits the caches, and the findBMU computation takes sufficiently long we achieve even better linear speedup.
The best achieved speedup between the compiler unoptimized version and fully parallel four thread version with
our optimizations and additional compiler optimizations is approximately 43 times. Comparing the compiler optimized single threaded version and our fully optimized four thread version, we achieved 11.6 times speedup.
In our future work we would like to enhance the simulation of the THSOM neural network to the Cell Broadband Engine processor.

Acknowledgments

This work has been supported by the internal grant of Czech Technical University in Prague, IG: CTU0812213 – “Hardware Acceleration for Computational Intelligence” and the research program ”Transdisciplinary Research in the Area of Biomedical Engineering II” (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

REFERENCES

SIMULATION MODELING OF A CLUSTER-TO-CLUSTER GRID

Lazar Kirchev  
Institute on Parallel Processing  
Bulgarian Academy of Sciences  
25A, Acad. G. Bonchev Str., 1113 Sofia, Bulgaria  
E-mail: likirchev@acad.bg

Vasil Georgiev  
Faculty of Mathematics and Informatics  
Sofia University “St. Kliment Ohridski”  
5, James Bourchier Blvd., 1164 Sofia, Bulgaria  
E-mail: v.georgiev@fmi.uni-sofia.bg

KEYWORDS  
Grid computing, Simulation Modeling, Queuing Models, Scalability, Performance Parameters.

ABSTRACT

This paper describes a queuing based simulation model of a clustered grid computing framework we have developed. Our grid architecture combines the advantages of the scalable and transparent wide-area distributed processing and the cluster-range centralized control. The grid level is implemented in a cluster-to-cluster model. In order to analyze the capabilities, and mainly the scalability, of our proposed architecture we defined a queuing model of the system and performed simulations based on the model. The results give us some important performance parameters as efficient cluster size, optimal intercluster throughput, etc.

INTRODUCTION

Grid computing has emerged as a potential next generation platform for solving large-scale problems in science, engineering, and public services. It is expected to involve millions of resources, scattered across multiple organizations and administrative domains. The management and scheduling of resources in such large-scale distributed systems is complex and therefore demands sophisticated tools for analyzing and fine-tuning the algorithms before applying them to the real systems. Simulation appears to be the only feasible way to analyze distributed systems of such a scale. It works without making the analysis mechanism unnecessary complex, by avoiding the overhead of coordination of real resources. Simulation is also effective in working with very large hypothetical problems that would otherwise require involvement of a large number of active users and resources, which is very hard to coordinate and build at large-scale research environment for investigation purpose.

In this paper we use simulation to analyze the properties and behavior of a grid system which we are designing and prototyping. We are particularly analyzing the speedup and efficiency of parallel applications executed in the platform. We have already built a prototype of the system, but we have tested it on a very small infrastructure, so we want to explore its scalability through simulation. The system is called GrOSD (Grid-aware Open Service Directory) (Kirchev et al. 2007) and has hybrid architecture. The system is built of “virtual clusters”, which are organized in a peer-to-peer manner. The organization within a cluster is centralized, but among the clusters in the system it is decentralized. This approach differs from the one in most systems, which implement either centralized approach (e.g. ALICE (Teo and Wang 2004)) or decentralized (e.g. H2O (Kurzyniec et al. 2003)). The hybrid approach provides for better scalability and management of the system. The centralization within the clusters facilitates their control, while the decentralization among the clusters improves the scalability of the whole system. The clusters are “virtual”, because they may contain machines which belong to different administrative domains, and in this sense they are a logical unit for organizing resources and users. One of the main purposes of the system is to be lightweight (not to use complex grid middleware such as Globus (Foster 2006.), for example), to have minimal requirements to the nodes, which are part of it and to be easy to use and administer.

The system is completely service-oriented, with everything in it represented by services. The system middleware consists of a number of system services, which perform the middleware functions, such as monitoring, scheduling, resource and user management, etc. Of each of these system services there runs a single instance in each cluster, eventually on one machine, although it is not necessarily so. One of the system services is special in that an instance of it runs on each node of the cluster – it makes the nodes part of the cluster and is responsible for the execution of tasks on the node. The programmatic units, which the users may use, are also represented by services. Some services are only visible in the cluster, where they are published, and thus may be called only by users of this cluster, while other services are exported – i.e., they are visible in the rest of the clusters in the system. The access to a cluster is portal-based and requests for services are received by the resource management service, which either sends them to a node in the cluster for execution, or forwards the request to another cluster, if the request is for a service, exported by another cluster. The grid system is general-purpose and is not specifically designed as a computational or data grid – it may perform one task or another depending on the services, which are deployed in it.

Different tools exist for simulation of distributed and grid systems. Following are descriptions of some of them. The GridSim toolkit (Sulisito et al. 2003) allows modeling and simulation of entities in parallel and distributed computing (PDC) systems – users, applications, resources, and resource brokers (schedulers). It provides a comprehensive facility for creating different classes of heterogeneous resources that can be aggregated using resource brokers. A resource can be a single processor or multi-processor with shared or distributed memory and

1 This work was supported by SUGrid (Project No. VU-MI-110/2005)
managed by time or space shared schedulers. The processing nodes within a resource can be heterogeneous in terms of processing capability, configuration, and availability. The resource brokers use scheduling algorithms or policies for mapping jobs to resources to optimize system or user objectives depending on their goals.

Beosim (Jones et al. 2004), is a discrete event driven simulator designed to model a mini-grid as a collection of (possibly heterogeneous) computational clusters connected via a dedicated interconnection network. Beosim can be driven via synthetic workload distributions that are characterized through the use of randomly generated arrival and service processes. BeoSim addresses the contention for bandwidth that simultaneously co-allocated jobs create within a grid's interconnection network. It is being used to design and study bandwidth-aware scheduling algorithms that improve job turnaround time by addressing degraded job run-time performance due to intercluster network bandwidth utilization.

GPSS World (Cox 1991) is a discrete event simulation environment implementing the GPSS Simulation language, which also provides continuous modeling capabilities. It is general purpose and allows the modeling and simulation of various types of systems, ranging from port and traffic light management to network traffic, complex scheduling algorithms and distributed systems. GPSS World extends the simulation language with a programming language, which facilitates the construction of simulations. While the first two tools provide a variety of features, they are very specialized, on one hand, and on the other do not match our simulation model's level of abstraction. That is why we chose to use GPSS World, as the most general-purpose of all, and as such giving most freedom for expressing different models and levels of abstraction.

**SIMULATION MODEL**

In order to study the parameters of the system when the participating clusters consist of a larger number of nodes we conducted simulation analysis. For this purpose we constructed an abstract simulation model, in which the GrOSD virtual clusters are considered as consisting of a cluster server, which receives user requests, and $n$ homogeneous working nodes, which serve the requests. Figure 1 depicts this abstract model.

![Abstract Model of a Virtual Cluster in the GrOSD System](image)

The cluster server receives several user request flows. These are as follows: requests from cluster users for services in the same cluster (intracluster requests); requests from cluster users for services, exported by other clusters, which will be forwarded for execution to another cluster; requests from users of other clusters for services, which the local cluster exports. The nodes in the cluster receive two request flows – tasks for execution, sent by the cluster server, and local tasks, which model the fact that the nodes are not dedicated to the cluster and are also used by local users with higher priority. The requests for services are served in an opportunistic mode – the low priority opportunistic requests are served when the local node is either idle, or with very small load.

The model is focused on the functioning of the separate virtual clusters and studies basic cluster parameters. The membership of the cluster in the grid is modeled by the flows of requests from and to the other clusters. The cluster server receives also one additional flow of requests, which is included in the model for completeness, and is not considered as part of the flows of requests for serving. This flow consists of requests sent from the nodes in the cluster when they finish performing a task. It models the events mechanism, used in the platform.

The service requests, coming to the cluster server are placed on a system queue, from which the server selects tasks for execution in a FIFO mode. The tasks for execution have equal priority. The length of the system queue is not limited. The serving of a request by the cluster server models the discovery of an appropriate node, to which the request will be sent for execution. The cluster server chooses the first free
node – i.e., which does not execute a cluster task at the moment.
The notifications for finished execution coming from the cluster nodes trigger selection of a new task for execution from the queue, since the notification guarantees the presence of a free node, to which the task may be sent. These notifications have higher priority and are not placed in the system queue, but are processed directly.
The tasks, sent by the cluster server to the nodes for execution, and the local user tasks in the nodes, are placed in a local system queue in the node. The local tasks have higher priority. Tasks for execution are selected on a FIFO basis, but the priority of the task is also considered.

SIMULATION MODEL PARAMETERS

The simulation model defines several basic parameters, which are used to describe the entities of the model and their relations.
The request flows, which enter the cluster server, have different incoming rates (intensities). The following rates are defined:
\[ \lambda_i = \frac{T_i}{t} \] incoming rate of the intracluster requests;
\[ \lambda_{inc} = \frac{T_{inc}}{t} \] incoming rate of requests from other clusters;
\[ \lambda_o = \frac{T_o}{t} \] incoming rate of requests for forwarding to other clusters. \( T_i \), \( T_{inc} \) and \( T_o \) are the number of the respective requests, received for time \( t \).
The cluster server serves the different types of requests with different serving rates. The serving rate is defined as the average number of requests of the respective type, which may be served for a time period and depends on the average processing time of the request. The following serving rates are defined:
\[ \alpha_i = \frac{1}{t_i} \] intracluster requests serving rate;
\[ \alpha_{inc} = \frac{1}{t_{inc}} \] serving rate of requests from other clusters;
\[ \alpha_o = \frac{1}{t_o} \] serving rate of requests for forwarding to other clusters. \( t_i \), \( t_{inc} \) and \( t_o \) are the average times for serving the respective request type.
An indicator of the cluster server status is its load factor, which is defined as
\[ \rho_s = \frac{\lambda_i}{\alpha_i} \] where \( \lambda_i \) is the aggregated incoming rate of requests from the different flows, and \( \alpha_i \) is the aggregated serving rate for the incoming requests. \( \lambda_i \) and \( \alpha_i \) are calculated with the formulas
\[ \lambda_i = \lambda_i + \lambda_{inc} + \lambda_o \]
\[ \alpha_i = \frac{\alpha_i \alpha_{inc} \alpha_o (\lambda_i + \lambda_{inc} + \lambda_o)}{\lambda_i \alpha_{inc} \alpha_o + \lambda_{inc} \alpha_o \alpha_i + \lambda_o \alpha_i \alpha_{inc}} \]
The parameter \( \rho_s \in [0,1] \) - if it is outside the interval the status of the server is undefined.
For each node \( i \) are defined the following parameters:
\( \lambda_{ci} \) - incoming rate of the tasks, sent to the node by the cluster server;
\( \lambda_{loc} \) - incoming rate of local tasks;
\[ \alpha_{ci} = \frac{1}{t_{ci}} \] serving rate of a cluster task, where \( t_{ci} \) is the average cluster task execution time;
\[ \alpha_{loc} = \frac{1}{t_{loc}} \] local tasks processing rate, where \( t_{loc} \) is the average local task execution time. For the nodes is also defined a node load factor
\[ \rho_i = \frac{\lambda_{ci}}{\alpha_{ci}} \] where \( \lambda_{ci} \) is the aggregated incoming rate of the two request flows, and \( \alpha_{ci} \) is the aggregated serving rate. \( \lambda_{ci} \) and \( \alpha_{ci} \) are calculated by the formulas
\[ \lambda_{ci} = \lambda_{ci} + \lambda_{loc} \]
\[ \alpha_{ci} = \frac{\alpha_{ci} \alpha_{loc} (\lambda_{ci} + \lambda_{loc})}{\lambda_{ci} \alpha_{loc} + \lambda_{loc} \alpha_{ci}} \] for \( \rho_i \) should be valid
\[ \rho_i \in [0,1] \].
For the nodes is also defined a coefficient for connectedness to the cluster as
\[ \eta_i = \frac{\lambda_{ci}}{\lambda_{loc}} \] it shows the ratio between the cluster and local tasks, executed by the node.
For the cluster as a whole is defined a coefficient for utilization of the cluster for cluster tasks as
\[ \phi = \frac{\lambda_i + \lambda_{inc}}{\sum_{i=1}^{n} \lambda_{loc}} \] it shows the ratio between the cluster tasks and the local tasks on all nodes in the cluster.

Table 1: Quantitative Expression of the Granularity of the Tasks

<table>
<thead>
<tr>
<th>quantitative measuring of the granularity</th>
<th>fine</th>
<th>medium</th>
<th>coarse</th>
<th>extracoarse</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0.01 \leq \gamma \leq 0.1 )</td>
<td>001</td>
<td>001</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>( 0.01 \leq \gamma \leq 0.05 )</td>
<td></td>
<td>001</td>
<td>001</td>
<td></td>
</tr>
<tr>
<td>( 0.01 \leq \gamma \leq 0.005 )</td>
<td></td>
<td></td>
<td>001</td>
<td></td>
</tr>
<tr>
<td>( \gamma \leq 0.005 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An important coefficient for the simulation analysis of the grid system is the granularity \( \gamma \) of the tasks, sent to the cluster server for execution. This coefficient is defined as the
ratio between the average task execution time and the average time which the cluster server spends to serve the request (discovering a node and sending the task):

\[ \gamma = \frac{t_e}{t_s} \]

where \( t_e \) is the average execution time, and \( t_s \) is the average serving time. The granularity is a synthetic parameter, whose quantitative measurement is very relative. In Table 1 we classify the granularity according to its coarseness.

We base our simulation on experiments with a prototype of the described grid system, and for some of the simulation parameters the values are obtained from these experiments. Such are the incoming request rates, the serving rates, the average execution and average serving times, which define the task granularity.

The task arrivals to the cluster server are considered to be independent and forming asynchronous computational load. The tasks sent from the cluster server to the nodes, as well as the local tasks, are also independent. The arrival of requests is assumed to have a Poisson distribution, as for example in (Jones et. al. 2004; Jones et. al. 2008).

In order to prove the adequacy of the simulation model we conducted simulation experiments and tests at one and the same conditions regarding the granularity of the tasks and the number of nodes. Figures 2 and 3 show the results for the speedup respectively for the tests and the simulations.

**SIMULATION RESULTS**

The simulation is implemented with GPSS World. Through simulations is studied the behavior of clusters with different sizes – from 1 to 256 nodes. Different values of the granularity and the node load factor are investigated. The clusters are considered homogeneous, because the speedup and the efficiency parameters, which are analyzed, are generally defined for homogeneous systems. The average serving time of a request for local services, coming from within the cluster, and from other clusters, is considered the same, because the requests differ only by their source. The serving time of a request, which will be forwarded to another cluster is larger than that of the other types of requests, because it is served in two phases – first a query is sent to the other clusters, and then it is forwarded to one of them. For calculating the speedup and efficiency of the cluster, the execution time of a particular number of requests is measured when the number of executing nodes varies. Figures 4 to 9 show the graphs for the speedup and the efficiency for three different values of the local node load factor \( \rho_l \). Each graph contains a family of four curves, which correspond to the different granularities from Table 1.

The simulation gives two basic results, which are very important for the analysis of the grid system. On one hand it shows which is the optimal granularity of the tasks, which are to be executed in the system, and on the other – the scalability of the system.

The diagrams show that for the different values of the local nodes load factor for the tasks with coarse and extracoarse granularity the system achieves the best speedup and efficiency.

---

Figure 3: Speedup in Simulations

Figure 2: Speedup in Tests
The other important result from the simulations is that the system is scalable. The results show, that the clusters scale to more than 100 nodes, guarding a linear speedup and acceptable efficiency - approximately 0.6. Such a scalability level is satisfactory for the system under development. The main idea of this C2C architecture is the formation of not very large peer clusters, the scaling of the system being done by adding new clusters rather than adding too many nodes to an existing cluster.

When the granularity is fine the impact of the load on the cluster server, caused by the serving of the requests, on the speedup and efficiency is rather obvious, thus hindering the high efficiency. The diagrams show that for fine granularity the efficiency is the lowest and the threshold at which the speedup stops increasing is reached for much smaller number of nodes, than for coarser granularities. Practically, the grid systems are oriented towards execution of tasks with coarse and extracourse granularity – usually whole programs (services) or whole workflows. From this point of view the results of the simulation are satisfactory, since in the general case the analyzed system will not be used for fine granularity tasks.

The analysis of the speedup and efficiency in relation to the different node load shows that under high load the speedup increases more slowly with the increase of the nodes number,
and the efficiency is lower. Since the node load actually depends on the number of nodes in the cluster, if the load becomes rather large, it may be off-loaded with the addition of new nodes. This approach may be used if the number of nodes does not exceed the optimal number of nodes in the cluster.

![Graph showing speedup and efficiency for $\rho_i = 0.9$.](image)

**Figure 8:** Speedup for $\rho_i = 0.9$

![Graph showing efficiency for $\rho_i = 0.9$.](image)

**Figure 9:** Efficiency for $\rho_i = 0.9$

**CONCLUSIONS AND FUTURE WORK**

In this paper we presented a C2C architecture for a grid system and described a simulation modeling approach for its performance analyses. An abstract model of the system is defined, which facilitates the simulations of the system. The model is described by a number of parameters. The simulations, based on this model, are implemented with GPSS World. They examine basic parameters of the designed system, such as scalability, optimal granularity of the tasks, behavior of the system at different load levels. One of the primary conclusions of this paper is that while clustering can dramatically reduce the system control overloaded and provide a good scheme for building lightweight grid architectures, its ability to do so depends heavily on the characteristics of the basic architecture parameters – mostly arriving workload stream, application services’ granularity, cluster size limitations and related performance of the servers. The results of the simulations showed us that the optimal cluster size is approximately 100 nodes, that the system is better suited for coarse and extracoarse tasks, that the for optimal efficiency of the cluster the load factor of its nodes should not be much more than 0.7.

We also plan on developing models to address the impact of intercluster system and application exchange. Other plans include simulation of a different system organization, envisioning hierarchical grid, in order to compare the results for the two types of architecture.

**REFERENCES**


MULTILAYERED AND HETEROGENEOUS MODELLING AND SIMULATION OF NATURAL COMPLEX SYSTEMS

Emilie Broutin
Paul Bisgambiglia
Jean-François Santucci

University of Corsica, SPE
CNRS, UMR N°6134
Campus Grimaldi
20250 CORTE

Email : {broutin | bisgambi | santucci}@univ-corse.fr

KEYWORDS
DEVS, Multilayer, HLA, reusability

ABSTRACT

DEVS formalism allows modeling and simulation of discrete event phenomenon. The actual frameworks and simulators only allow taking into account models which are dealing with data at the same time scale and with the same level of abstraction. It means that data must be on the same units and formats. Without this standardization of the data, data exchange is not possible between different models.

In this paper we present a new way for simulation of multilayered naturals systems. Our main goal is to set up an architecture that can contain all the different views of a natural phenomenon. For example in the case of catchment modeling we can integrate the snow model, the rain model and the catchment model in an unique model. With today simulation tools the only way to share data between different models need a central person in order to rewrite models and make sure the outputs of one of the models are in an adequate format corresponding to the associated inputs of the connected others models. Within the proposed architecture, models can communicate and share data without external intervention.

INTRODUCTION

The modelling and simulation of complex systems is a task requiring a lot of time to be achieved. One way to facilitate this task is to develop methodologies for creating and using highly inter-connectable components. Furthermore computer simulations have been becoming increasingly complex and require efficient system simulation framework [2]. The DEVS formalism defined by Prof. B.P. Zeigler in the late seventies has been emerged in the nineties as a standardized formalism for performing simulation of complex systems [12,13,14,15]. Usually design tools are associated with libraries of reusable modelling components that will make the description of models and also their validation much easier. Recently a set of research work has been oriented towards this direction. We can highlight the coupling of the DEVS formalism with two technologies allowing the re-use of already defined components and the distribution of the simulation process on several computers: (i) HLA – High Level Architecture [1,4,5,8,10,11] (ii) SOA:-Service Oriented Architecture [6]. However in both case the only way to interconnect components is to re-use components having the same type of data as input or output and whose behaviour has been described according to the same level of temporality.

In this paper we propose a new approach allowing to facilitate the re-use of components by taking into account components whose : (i) the input/output data can belong to different levels of abstraction and (ii) the occurrence time unit of events coming out or arriving on the component can be totally different from each other. In order to fully understand the interest of such an approach for the modelling and simulation process we present the three kinds of persons who are involved in the modelling and simulation of complex systems : (i) the Framework Builder, who is familiar with the internal structures of a generic software framework allowing to perform the simulation of complex systems ; (ii) the Application Builder whose role is to define appropriate models for a given class of applications and establish problem resolution strategies adapt to his class of applications by using the generic software framework defined by the Framework Builder ; (iii) the Final User whose role is to supply the problem description and other relevant information to the application system defined by the Application Builder.

In this paper we describe how to define a generic software framework allowing an Application Builder to interconnect DEVS components dealing with data at different levels of abstraction and different levels of temporality.

The DEVS formalism is a simple way in order to characterize how discrete event simulation languages may specify discrete event system parameters. It is more than just a means of constructing simulation models. It
provides a formal representation of discrete event systems capable of mathematical manipulation just as differential equations serve this role. Furthermore, by allowing an explicit separation between the modelled phase and simulation phase (Zeigler and al. 2000), DEVs formalism is one of the best ways of performing an efficient simulation of complex systems using a computer.

In the DEVs formalism, one must specify: 1) basic models from which larger ones are built, and 2) how these models are connected together in hierarchical fashion. Basic models (called atomic models) are defined by the following structure:

\[ CA = < X, S, Y, \delta_{in}, \delta_{ext}, \lambda, \tau_0 > \text{ where : } (i) \ X \text{ is the set of input values} ; (ii) \ S \text{ is the set of sequential states} ; (iii) \ Y \text{ is the set of output values} ; (iv) \ \delta_{in} \text{ is the internal transition function dictating state transitions due to internal events} ; (v) \ \delta_{ext} \text{ is the external transition function dictating state transitions due to external input events} ; (vi) \ \lambda \text{ is the output function generating external events at the output} , \text{ and} (vii) \ \tau_0 \text{ is the time-advance function which allows to associate a life time to a given state.} \]

The behaviour of an atomic model is illustrated as follows: the external transition function describes how the system changes state in response to an input. When an input is applied to the system, it is said that an external event has occurred. The next state \( s' \) is then calculated according to the current state \( s \). The internal transition function describes the autonomous (or internal) behaviour of the system. When the system changes state autonomously, an internal event is said to have occurred. The next state \( s' \) is therefore calculated only according to the current state \( s \). The output function generates the outputs of the system when an internal transition occurs. The time advance function determines the amount of time that must elapse before the next internal event occurs, assuming that no input arrives in the interim.

An atomic model enables us to specify the behaviour of a basic element of a given system. Connections between different atomic models can be performed by a coupled model (CM) (Zeigler 1976, Zeigler 1984).

A coupled model indicates how to couple (connect) several component models together to form a new model. This latter model can itself be employed as a component of a larger coupled model, thus giving rise to hierarchical construction. A simulator is associated with the DEVs formalism in order to execute a coupled model's instructions so as to actually generate its behaviour. The architecture of a DEVs simulation system is derived from the abstract simulator concepts (Zeigler and al. 2000) associated with the hierarchical and modular DEVs formalism. The abstract simulator allows the definition of a simulation tree whose root element is dedicated to the time advance management. We propose in this paper an approach allowing both to interconnect components whose input and output data are not at the same level of abstraction and to perform simulations of events whose occurrence date are not at the same level of granularity.

The proposed approach leans on the definition of a DEVs coupled model which is called in the rest of the paper the “Assembly Layer”. The objectives of this so-called “Assembly” layer are manifolds: (1) it will allow storing the set of data required when dealing with a given application, each data being stored under a given format; (2) it will allow each component of a given application to be interconnected even if their respectively processed data are not under the same format; (3) it will allow each component to be simulated at different levels of temporality; (4) it will allow to drive the simulation process as a root element of the abstract simulator tree defined by B.P. Zeigler.

The paper is organized as follows. Section 2 presents in detail the specifications of the Assembly coupled model required in order to be able to deal with DEVs simulation of components at different levels of abstraction or temporality. The implementation of the “Assembly” coupled model within the DEVs formalism is given in section 3. We will point out the atomic models we have defined in order to satisfy the four previously mentioned objectives. Finally section 4 concludes this paper and provides some perspectives of work.

SPECIFICATION OF THE ARCHITECTURE

Layers

The actors involved in the modelling of a complex system may come from different domains. The idea is to set up a software architecture allowing the models from different fields to communicate whatever the abstraction level or time level. We call “layer” a model belonging to a given domain which describes a part of a studied phenomenon. Classical approaches incorporate all the data at the same level. These approaches lead to two main drawbacks: (i) the resulting models are too specific and non reusable or (ii) one must define for all models a common base for abstraction and time level. In both cases the characteristics of the data that will be exchanged between the models involved in order to studied a given phenomenon have to be defined in an identical manner for all the models. So it reduces the level of reusability of the models.

We propose in this paper to define a software architecture which can incorporate all kind of models whatever the characteristic of the data which are to be exchanged between models involved when studying a given phenomenon.

In order to reach this objective we have defined two kinds of layer:

- “Behavioural layers” which represent the system’s dynamic or a part of the system’s dynamic. A behaviour layer is a model allowing to describe part of the behaviour of the studied phenomenon. Generally more than one behavioural layer is required in order to fully study a given phenomenon.
- “Assembly layers” which allow the communication between behavioural layers. Own to this kind of layer, models can send and receive data from and to behavioural layer. So every model can share data with
his external environment.

- There is only one assembly layer in the architecture.
- The assembly layer allows to establish the link between the behavioural layers.
- All the data exchange between the behavioural layers must transit by the assembly layer.
- The assembly layer is in charge of the time management during the simulation.
- Behavioural layers must subscribe to the assembly layer for sending or receiving data.
- Behavioural layers must declare the type of data that they send or want to receive (time units, format...).
- The Assembly layer must perform the conversion operations needed for transferring data in a comprehensible format for the behavioural layer which has subscribed.

Section 3 will present how we have implemented the “assembly layer” in the DEVS formalism.

**SPECIFICATION IN DEVS FORMALISM**

This section presents the functioning of the assembly layer and its representation in the DEVS formalism.

We have seen above that time management has to be performed by the assembly layer. Usually in the DEVS formalism a ROOT component of the abstract simulation tree corresponding to a behavioural model (atomic or coupled model) is in charge of the time management.

In the proposed architecture, the ROOT component of each behavioural layer must refer to the assembly layer to obtain the acknowledgement for time advancement. This acknowledgment is given if and only if all the behavioural layers managed by the assembly layer are authorized to advance their clock. That is to say when all of them have asked for an acknowledgment for time advancement. With this method we avoid a possible loss of data that may occur: (i) if a behavioural layer requires advancement in time faster that the other behavioural layers involved in the study of a given phenomenon; (ii) when a data is updated by a behavioural layer and that this update will involve an event to be performed on a layer which is already at a higher step of time.

The assembly layer is representing by a DEVS coupled model (see figure 2).

We can observe in figure 2 that each behavioural layer is linking to an atomic model of the “assembly layer”. Each kind of data is stored in a recording media that is identified by a state. The atomic model which is linked to a given behavioural layer may point to several recording medias.

Depending of its state, the atomic model will choose the recording media in which the required data is stored. Behavioural layers must first declare what type of data they would receive and what kind of data they want to share with the other behavioural layers. This modelling feature allows the assembly layer to perform the correct links between the atomic models and the recording media. Data are stored with a time-stamped method.

When a behavioural layer wants to advance its clock, it...
must send a Tcci to the assembly layer. The time management is performed by the assembly layer when all the behavioural layers have send their time advance require (that implies that there is no event to be operated in all of the behaviour layers before Min(Tcci)). In this case the assembly layer can acknowledge the advancement of the time at the following value: minimum value between Min(Tcci) and the minimum of the littlest Time-stamped event stored in each recording media. Time is set at this new value and all of the events stored in the recording media are sending to the appropriated behavioural layer.

CONCLUSIONS AND PERSPECTIVES

We have presented in this paper a first approach in order to model and simulate interconnections of DEVS components involving different levels of abstraction and different levels of granularity. Our approach leans on the definition of a DEVS coupled model called “Assembly layer” which allows to perform simulation of DEVS models whose input and output are not compatible in terms of both domain definition and simulation time. After having listed the precise specification of such a coupled model, we have detailed the DEVS implementation of this component by describing the atomic models which are composing such coupled model as well as the way they are interconnected. The next step of our work concerns the software implementation of this “Assembly coupled model” and the validation of the resulting software on concrete applications. In order to validate the proposed approach we choose to implement the described atomic and coupled models using the PythonDEVS simulator [7,3]. The Python-DEVS Modeling and Simulation package provides an implementation of the standard classic DEVS formalism described in section 1. The package consists of two files, DEVS.py and simulator.py. The first one provides class architecture that allows hierarchical classic DEVS models to be easily defined by subclassing the AtomicDEVS and CoupledDEVS classes. The Simulator engine (SE) is implemented in the second file. Based on the principles of simulation describe in section 1, it allows to perform discrete event simulation. Even if the PythonDEVS software involves a simulation engine which offers limited means to terminate a simulation and provides no easy model-reinitialisation possibilities we have been able to use it in order to efficiently test our approach. We shall be specially able to easily introduce the “Assembly coupled model” as described in section 3 because of the open source quality of the PythonDEVS package by subclassing both the atomic model Class and the coupled model class inherent to the PythonDEVS package. However in order to be able to perform the simulation algorithm of PythonDEVS we have to transform the simulation file called simulator.py. So our next step will consists in: (i) carefully defining the algorithm allowing the “assembly coupled model” to be considered as a root model; (ii) modifying the python
software implementation of the simulation process of the simulator.py file of the PythonDEVS package in order to perform simulations of models at different levels of temporal granularity and to integrate models at different levels of abstraction. A last step will consists in validating the resulting software on a concrete application issued from natural complex systems.

REFERENCES

[10] Zacharewicz, G. “Environnement GDEVS compatible HLA”.

BIOGRAPHIES

Emilie Broutin is a PhD student in computer science in the University of Corsica. Her current research focuses on DEVS multilayered modeling and simulation.

Jean-Francois Santucci is Full Professor in Computer Sciences at the University of Corsica since 1996. His main research interest is Modelling and Simulation of complex systems. He has been author or co-author of more than 100 papers published in international journals or conference proceedings. He has been the scientific manager of several research projects corresponding to European or industrial contracts. Furthermore he has been the advisor or co-advisor of more than 20 PhD students and since 1998 he has been involved in the organization of more than 10 international conferences. He is conducting newly interdisciplinary researches involving computer sciences, archaeology and anthropology: in the one hand he is performing researches in the archeoastronomy field (investigating various aspects of cultural astronomy throughout Corsica and Algeria using tools issued from Computer Sciences) and on the other hand he is applying computer sciences approaches such as GIS (Geographic Information Systems) or DEVS (Discrete EVent System specification) to anthropology.

Paul Bigambiglia is a professor in Computing Sciences at the University of Corsica. He is responsible of the modeling and simulation team of the UMR CNRS 6134. His research activities concern the techniques of modeling and simulation of complex systems and the test of systems described at high level of abstraction. He makes his researches in the laboratory of the UMR CNRS 6134.
ANALYSIS OF TEMPORAL DEPENDENCIES OF PERCEPTIONS AND INFLUENCES FOR THE DISTRIBUTED EXECUTION OF AGENT-ORIENTED SIMULATIONS

Nicolas Sébastien
Rémy Courdier
Didier Hoareau
Marc-Philippe Huget

*EA255 LIM-IREMIA
2 rue Joseph Wetzel
97490 Sainte-CLOTilde
France
nicolas.sebastien
remy.courdier @univ-reunion.fr
didier.hoareau

**LISTIC
5 chemin de Bellevue
BP 80439
74944 Annecy-Le-Vieux cedex
France
marc-philippe.huget@univ-savoie.fr

KEYWORDS
Multiagent simulation, Temporality model, distributed parallel execution

ABSTRACT

Agent-Oriented Simulations (AOS) are helpful to understand real complex systems particularly via their modeling ability and the observations they provide. To fully exploit those simulations, AOS platforms should offer shorter simulation time even if simulated systems are more and more complex.

In this paper, we propose a method to minimize the execution time of AOS through the parallel execution of simulation agents. Our solution is ensured by a network of AOS platforms that relies on the identification of agents’ temporal dependencies.

Our proposal is based on the Temporality model that allows the agents to express their temporal dynamics.

INTRODUCTION

Agent-Oriented Simulation (AOS) is an appealing approach to model complex systems. Reproducing the real system complexity, AOS provides multi-scale observation of a model (Ralambondrainy and Courdier 2007) in various configurations. This approach faces a scalability problem with ever more complex simulations implying throng of interacting agents which increase simulation time.

Grid technologies (Buyya and Venugopal 2005, Foster 2002) tackle such scalability problems. However, distributed simulation requires a specific scheduler to maintain time causality between events generated on different computing devices. Based on a temporal model (commonly constant step, event-driven or mixed approaches), the scheduler impact the overall simulation time.

We recently developed a temporal model named the Temporality model. This one supports various classical temporal models like constant step or event-driven approaches, by adding informations to agents’ activation timestamps. In this paper, we show how to exploit these informations to propose a scheduler based on the Temporality model that provides parallel execution of agent-oriented simulations.

This paper is structured as follows. We first define our scheduling algorithm based on the Temporality model before detailing how to detect causality links established in the simulated model. We show how to provide parallel execution of simulation given the causality links before we conclude this paper.

SCHEDULING AND PARALLELISM

The Temporality model

The Temporality model is defined as a temporal model for Agent-Oriented Simulations. One of its major assets is the fact that it eases the identification in time of the different dynamics agents participate in. Indeed, when using the Temporality model, agents define their own activation times. The scheduler is so driven by the needs of the agents composing the simulated model. Such needs are described by a data structure called temporality and defined by $t = id, d, f, p, v$ with:

- $id$: temporality identifier. Such identifier can be used by agents to associate a specific behavior with the temporality.
- \([d, f]\): time interval during which the temporality can be triggered

- \(p\): the time interval between two executions of this temporality \((p = 0\) if the action is executed once).  

- \(v\): the variability that defines the temporal precision for the triggering of a temporality.

A temporality trigger the agent’s behavior at each date \(x\) verifying: \(x = d + p \times k\), with \(k\) an integer such as \(0 \leq k \leq n\) and \(n\) the biggest integer verifying \((d + p \times n) = f\). Agents first give the scheduler their temporalities at simulation initialization. Then, during simulation, agents are free to modify their temporalities as part of their behavior: this enables them to adapt their own dynamics to the situation.

Considering the temporal engine, we introduce two additional data structures:

- **Time Slot**: a point on the simulated temporal axis on which the simulator will have to trigger the execution of an agent’s behavior.

- **Tempo**: a structure that wraps temporalities that have the same period and are “situated” on the same temporal slot. The period characterizes the tempo as all the temporalities it contains will have the same rhythm.

### Temporality model scheduling

At simulation start, we consider that each is agent is executed on a node and will remain on that node during simulation. We also consider that the scheduler is centralized; this latter makes the simulated time progressing by jumping from one time slot to another. At each time slot, each tempo is executed by triggering all the temporalities it is composed of. When all simulation nodes have finished to execute a tempo, the scheduler determines the new time slot to attach the considered tempo to by considering its period. Thus, it only requires one operation to determine the next triggering date of each temporality composing the tempo.

To avoid inconsistency, environment remains unchanged during a time slot execution. Execution nodes must then diffuse environment updates on the network before processing the next time slot. After this synchronization, the scheduler trigger execution of the next time slot and so on until simulation end.

As agents declare their temporalities at the beginning of the simulation, it is possible to obtain a first draft of the simulated temporal axis. The Figure 1 shows a simulated temporal axis followed by the scheduler for the distributed execution: agents situated on different simulation nodes have different temporalities colors.

![Figure 1: Simulated temporal axis of the Temporality model](image)

### parallel scheduling

We aim at proposing a scheduler that supports parallel and distributed execution of AOS based on the Temporality model. Our main concern is to minimize the global simulation time. We use the temporal information to define a scheduler which does not depend on the application’s semantic.

Such a scheduler enables the triggering of tempos at the most early real time \((i.e.\) the time we live in) while ensuring that this early execution will not introduce inconsistency in the simulated model. For example, considering a resources producer \(P\) and a consumer \(C\) in a model which states the production before the consumption, an inconsistency would be having \(C\) executed before \(P\) produces the resources.

Defining a scheduler which ensures both performances in simulation and coherence of the distributed model is an issue referring to the causality principle in distributed systems. The causality principle states that if a phenomenon (called cause) produces an other phenomenon (called effect) then the effect can not occur before the cause. Given this definition, the execution of an agent at the most early real time can be expressed by:

“An agent can be activated by the scheduler as soon as this one can ensure that there is no causality relation between this agent and other agents that have not already been activated and that precede it on the simulated temporal axis.”

In the field of distributed simulation, there are two general approaches that maintain the causality between events (Ferscha and Tripathi 1994). The conservative approach is based on the identification of causality links to avoid a violation of this principle. The optimistic approach allows a causality violation but uses different mechanisms to recover such errors: it restores a previous state to take into account the event that caused the violation.

In this paper, we propose a scheduler based on the conservative approach. This choice leads us to identify the causality links that exist between agents.
IDENTIFICATION OF CAUSALITY LINKS USING THE TEMPORALITY MODEL

Definition of a causality link based on our simulation model

In our model, agents interact through the environment. They perceive their environment and influence it in order to produce an alteration in it (Ferber 1999). An agent a is influenced by an agent b if the perception made by a differs whether b has been activated before or not. A difference in a perceptions means that b dropped an influence that modified the perceived world of a. The perception/influence mechanisms establish causality links between agents. As perceptions and influences are made during agents activation time (triggered by temporalities), identifying agents causality links consists in identifying causality links between temporalities.

Causality links research

Causality links between temporalities come from two facts:

i. Both two considered temporalities are defined by the same agent. Behaviors associated to these temporalities uses the same agent’s data.

ii. The agent activated by the temporality which is most advanced on the simulated temporal axis (we will call it later temporality) perceives a part of the environment that have been influenced by an agent activated by a previous temporality (the earlier one).

The fact (i) denotes a causality link that exists between two temporalities of an agent. The behavior associated with the earlier temporality can update the agent internal data: behavior associated with the later temporality is so based on the potentially updated data. As we cannot easily determine the impact of the execution of a temporality on an agent internal data, we extend the fact (i) by considering that a causality link exists between two temporalities if these two temporalities belong to the same agent.

The fact (ii) establishes causality links between temporalities through the environment. An agent, during the execution of a temporality t, can produce an influence that would lead to a modification of the environment. All the temporalities during which an agent behavior would be affected by this influence present a causality link with t.

To identify such causality links, it would require to know about the scope of each influence and perception of each agent. Such information is a priori hard to determine if the users are not asked for it during the conception phase. This is why we propose to enlarge the conditions of causality links detection induced by the fact (ii) in considering the multi-environment approach (Payet et al. 2006).

The multi-environment approach consists in describing the world agents evolve in by a set of environments, each environment having its own evolution laws. An agent can perceive and influence several environments during a temporality. With this approach, we will make considerations on environments on which perceptions and influences are made: we consider that each request can impact the whole environment it is made on.

By using the multi-environment approach, we can replace the fact (ii) with:

“There exists a causality link between two temporalities if the agent activated by the later temporality makes a perception on an environment that have been influenced by the agent activated by the earlier temporality.”

Formalization

Using the Temporality model and the multi-environment approach, we consider two facts to identify causality links:

i. Both the two temporalities have been defined by the same agent.

ii. The agent activated by the later temporality does a perception on an environment that have been influenced by the agent activated by the earlier temporality.”

Causality links induced by the fact (i) are trivial to identify.

The fact (ii) requires information about agents’ behavior: we have to know the environments that will potentially be perceived and the environments that will potentially be influenced during the temporality’s activation.

We define $E = \{E_1, E_2, ..., E_n\}$ the set of environments described in the simulated model, we call:

- $E_R(a, t)$ the set of environments that can be perceived by agent $a$ at the temporality $t$ ($R$ stands for read).
- $E_W(a, t)$ the set of environments that can be influenced by agent $a$ at the temporality $t$ ($W$ stands for written).

By using the previous notations, we propose an extension of the description of temporalities $t = \{id, d, f, p, v, E_R(a, t), E_W(a, t)\}$.

We call $t_{sl}$ the simulated timestamp of the current activation of temporality $t$. We define $t_{sg}$ the global simulated timestamp of the system. At a given real time, $t_{sg}$ is the maximal simulated timestamp value for which all temporalities $t_i$, with $t_{sl} \leq t_{sg}$, have been executed.
Let $A$ the set of simulated model’s agents, a temporality can be executed according to the fact (ii) when $t_{sg}$ is such as the following relation is verified:

$$E_R(a, t_i) \cap \bigcup_{(k \in A, t_j \in \{t_{sk}, t_{sg} < t_{sk} < t_{si}\})} E_W(k, t_j) = \emptyset$$

We underline that the set $\{t_k, t_{sg} < t_{sk} < t_{si}\}$ is the set of temporalities that have not been yet executed and having a triggering time lower that the one of the considered temporality.

As both the Temporality model and the multi-environment approach aim at uncoupling real system’s dynamics, an agent will interact with few environments during a temporality. This tend to minimize $\text{card}(E_R(a, t))$ and $\text{card}(E_W(a, t))$ in these inequalities :

- $\forall a \in A, \forall t \in T E(a), \text{card}(E_R(a, t)) \leq \text{card}(E)$
- $\forall a \in A, \forall t \in T E(a), \text{card}(E_W(a, t)) \leq \text{card}(E)$

With $T E(a)$ the set of temporalities that have been defined by the agent $a$.

**SCHEDULER’S OPERATION**

Large-scale simulation implies throng of interacting agents and so a great number of temporalities. As considering temporalities would then require too much resources, we transpose our work on tempos. We so define, for the $T$ tempo:

- $E_R(T) = \bigcup_{t_k \in T} E_R(a, t_k)$ the set of environments that can be perceived during the execution of $T$.
- $E_W(T) = \bigcup_{t_k \in T} E_W(a, t_k)$ the set of environments that can be influenced during the execution of $T$.

With such sets, the fact (ii) consists in waiting for the $t_{sg}$ that verifies

$$E_R(T_i) \cap \bigcup_{(T_j \in \{T_{sk}, t_{sg} < t_{sk} < t_{si}\})} E_W(T_j) = \emptyset$$

Providing agents do not modify their temporalities, it is possible to determine a global simulation period, called $P$ : it is the least common multiple of each temporality’s period. We can then establish a dependency table between tempos on a global simulation period. During simulation, a modification of an agent’s temporalities will trigger a revision of the dependency table.

**Dependency table**

The dependency table is established considering the tuples $(E_R(T)|E_W(T))$ of each tempo. We propose the following algorithm:

```
for each tempo Tk do
    for each perceived environment El during Tk execution do
        identify tempos Ti such as tsi in [tsk-P, tsk] and during which El is influenced.
end for
```

This algorithm is used to determine causality links between tempos through the environment. The dependency table is then completed by integrating the causality links between tempos containing temporalities belonging to the same agents.

We consider the temporal axis of figure 2 on which we did not represent the tempos but only indicated the tuples $(E_R(T)|E_W(T))$. We consider, in this example, that each agent has declared only one temporality (there are no causality links induced by the fact (i)).

```
<env1env2> <env2env1> <env1env1> <env1env2>
T0  T1  T2  T3  T4  T0

<env1env1> <env2env2> <env2env1> <env1env1>
T0  T1  T2  T3  T4  T0
```

Figure 2: Example of asynchronous execution

The nature of the two environments here considered does not matter. Indeed, the mechanisms we previously presented does not take in consideration any knowledge on the type of environment. We deduce the dependency table from the example’s temporal axis (dependency is noted "→"):

- $T_0 \rightarrow T_2(P - 1) \land T_3(P - 1) \land T_4(P - 1)$
- $T_1 \rightarrow T_2(P - 1) \land T_3(P - 1) \land T_4(P - 1) \land T_0$
- $T_1 \rightarrow T_2 \land T_3 \land T_4 \land T_0(P - 1)$
- $T_1 \rightarrow T_2\land T_3 \land T_4 \land T_0$

The notation $(P-1)$ means that the tempo depends on a tempo belonging to the previous global period: in figure 2, $T_0$’s execution depends on the execution of $T_2$, $T_3$ and $T_4$ at the previous period.

The dependency table is valid as long as no agent modify one of its temporalities. Manipulation of temporalities by agents is critical for the scheduler. Let us consider an example: the scheduler triggers a temporality $t_2$ during which the associated agent may perceive the environment $E_1$. Another temporality, $t_1$, have to be executed with $t_{s1} < t_{s2}$ but during which its associated agent will not influence $E_1$. The trouble is that during the execution of the behavior associated with $t_1$, the activated agent may declare a new temporality $t_3$ with $t_{s3} < t_{s2}$ and during which it would influence $E_1$. Such a situation would produce a causality violation as $t_3$ should be activated before $t_2$. The detection issue induced by temporalities modification can be resolved by determining the temporalities which activation may lead to a modification of the associated agent’s temporalities. In our approach, the Temporality model is represented as a temporal environment. Thus, agents declare their temporalities by
influencing this environment. A temporality activation is seen as a perception in the temporal environment.

With such an approach, the issue induced by temporalities modification is transposed into the detection of causality links through environments. The causality links that can be created by modification of temporalities are identified in exactly the same way as for other environments. So, the dependency table naturally takes into account perceptions/influences on the temporal environment.

**Scheduling in simulation**

We consider here a scheduler that supports distributed and parallel execution. With their temporalities distributed on the execution infrastructure, temps are executed on several nodes. Each node broadcast a message when it completes a tempo execution. When all platforms complete a tempo execution, this tempo is declared “accomplished” in the dependencies of the dependency table. Every tempo for which all dependencies are “accomplished” can be executed.

We detail how the scheduler works in the previous example of Figure 2. A light gray cell color signifies that the tempo is accomplished. The dark grey cell in the first column denotes the temps that are in execution. When the simulation begins, all temps that should be executed at “the previous period”, that are annotated $(P-1)$, are considered accomplished.

At $t_{sg} = 0$,

<table>
<thead>
<tr>
<th>$T_0$</th>
<th>$T_2(P-1)$</th>
<th>$T_3(P-1)$</th>
<th>$T_4(P-1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>$T_0$</td>
<td>$T_2(P-1)$</td>
<td>$T_3(P-1)$</td>
</tr>
<tr>
<td>$T_2$</td>
<td>$T_1$</td>
<td>$T_3(P-1)$</td>
<td>$T_4(P-1)$</td>
</tr>
<tr>
<td>$T_3$</td>
<td>$T_0$</td>
<td>$T_1$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>$T_4$</td>
<td>$T_1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At the end of $T_0$ execution on each node, $T_0$ is declared “accomplished” in the dependency table. As all dependencies of $T_1$ are satisfied, $T_1$ is executed. After its execution, both $T_2$ and $T_3$ can be executed. We so have the following situation:

At $t_{sg} = t_{s1}$,

<table>
<thead>
<tr>
<th>$T_0$</th>
<th>$T_2(P-1)$</th>
<th>$T_3(P-1)$</th>
<th>$T_4(P-1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>$T_0$</td>
<td>$T_2(P-1)$</td>
<td>$T_3(P-1)$</td>
</tr>
<tr>
<td>$T_2$</td>
<td>$T_1$</td>
<td>$T_3(P-1)$</td>
<td>$T_4(P-1)$</td>
</tr>
<tr>
<td>$T_3$</td>
<td>$T_0$</td>
<td>$T_1$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>$T_4$</td>
<td>$T_1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At the end of $T_2$ and $T_4$ execution, they become accomplished in the dependency table. We underline that $T_2(P-1)$ and $T_4(P-1)$ are also accomplished. $T_3$ can then be executed. When $T_3(P-1)$ becomes accomplished, the dependency table will be the same as for $t_{sg} = 0$ as a simulation period has been executed: another period can be executed. In this example, we have seen that the tempo $T_3$ can be executed in parallel with the tempo $T_2$.

**CONCLUSION AND FUTURE WORKS**

In this paper, we proposed a scheduler for the parallel execution of distributed agent-oriented simulations. Our proposal is based on the Temporality model to support different classical temporal models. The parallelization strategy is conservative: observing the perception and influence mechanisms, the agents’ causality links are identified to avoid a violation of the causality principle. We exploit the multi-environment approach and the information associated with temporalities to establish a dependency table between the agents’ activations. This table is used during simulation to enable the earlier-in-the-real-time execution of agents.

Our proposition takes advantages of the simulated model parallelism to aim at better support large-scale systems. Toward this goal, our formalism does not depend on the type of parallel architecture used: we can exploit all multi-processors architectures (supercomputers or network distribution).

Like most of works on parallel execution of simulations, the performances of the scheduler we presented here strongly depends on the simulated model (e.g. on interactions between agents, temporalities that have been defined, etc.). To take better advantages of simulations characteristics, we are working on an extension of our formalism to support the definition of strategies integrating the simulation’s dynamic aspect. A first research track would be to exploit the temporal information to anticipate agents reparation on the execution infrastructure.

**ACKNOWLEDGEMENTS**

This publication was made possible by a grant from the regional council of Reunion Island.

**REFERENCES**


ON THE DEVELOPMENT OF OBJECT ORIENTED PARALLEL PENTA-DIAGONAL PRECONDITIONED KRYLOV SUBSPACE (K-S) SOLVERS

Bipin Kumar, Martin Crane
School of Computing, Dublin City University
Glasevin, Dublin 9, Ireland
Email: {bkumar, mcrane}@computing.dcu.ie

Yan Delaure
School of Mechanical Engineering
Dublin City University
Glasevin, Dublin 9, Ireland
Email: yan.delaure@dcu.ie

Keywords: Parallel Preconditioners, Krylov-Subspace (K-S) Solvers, Load Balancing, MPI-C++.

ABSTRACT

Object oriented parallel penta-diagonal preconditioning techniques for Krylov-Subspace (K-S) solvers have been developed. The ILU, SSOR and D8 are used as preconditioners. Further, penta-diagonal versions of non-preconditioned and preconditioned CGM solvers using novel preconditioners have been developed. These solvers are applied to solve the linear system of equations obtained from discretization of 2D Laplacian equation. A New load balance scheme is designed for data distribution. Code have been developed using MPI-C++ and implemented on a Linux cluster. Speedup data are generated and compared for preconditioned and non-preconditioned CGM solvers. The ILU preconditioner gives faster convergence and better speedup when compared with the other two preconditioners. It has been observed that the ILU is the most suitable preconditioner for the parallel penta-diagonal CGM solver.

INTRODUCTION

K-S solvers [1, 15, 16, 17] are one of the most efficient iterative methods to solve the large linear system of equations resulting from discretization of Partial Differential Equations (PDEs) representing the multiphase flow problems [5, 11]. Large systems of equations usually rule out the consideration of direct solution methods due to prohibitive memory requirements [10]. Some popular K-S methods are Conjugate Gradient Methods (CGM), Bi-CGM, Conjugate Gradient Stabilized (CGSTAB), Bi-CGSTAB and GMRES [2, 19]. Discretisation of PDEs may result in non-symmetric and ill-conditioned sparse matrices. K-S methods can converge slowly when applied [7] to those ill-conditioned matrices. Preconditioning techniques [8, 9, 13, 18, 20] can improve the rate of convergence and robustness of the iterative solvers. Among other preconditioners applied to K-S solvers, Incomplete L-U (ILU) factorization, Incomplete Cholesky factorization (ICH) Symmetric Successive Over Relaxation (SSOR) and Diagonal Scaling (DS) [1, 4, 16, 19] are the most popular. In this paper, a novel penta-diagonal version of ILU, ICH, SSOR preconditioner and preconditioned CGM solvers have been developed. Large systems of equations call for the involvement of parallel iterative solvers.

In this work, systems of equations obtained from discretization of the Laplacian equation are solved by K-S solvers. A novel parallel penta-diagonal version of K-S solvers and preconditioners are developed. Depending upon the complexity and the application space, problems in the computational domain are continuously increasing in computational intensity. Days are required for the numerical simulation [7, 9] of problems from CFD, in particular multiphase flow problems [11]. This requires involvement of parallel algorithms for K-S solvers [7, 8].

This paper is organized as follows: in the next section the K-S solvers and matrix data structures are described. In the concomitant section the newly developed penta-diagonal preconditioners are described. Parallelization of K-S the solver is discussed in subsequent section followed by the details of parallelization of penta-diagonal preconditioners for Object Oriented Parallel Preconditioned (OOPP) solvers. Results are presented in the succeeding section. The last section deals with the observations made indicating the future trends of research.

K-S SOLVERS

Discretization of PDEs converts them into an algebraic system of equations represented by \( A\mathbf{x} = \mathbf{b} \) where \( A \) is \( n \times n \) sparse matrix and \( \mathbf{x} \) and \( \mathbf{b} \) are \( n \) dimensional vectors. The need for a special data structure is called for to store a sparse matrix in the computers due to time as well as space(memory) constraints. Literature [10] reveals several common sparse matrix formats. Among these are Compressed Sparse Column (CSC), Compressed Sparse Row (CSR), index format and diag-
Penta-diagonal Preconditioners

The algorithm of dense ILU preconditioner as illustrated in Fig.(2(b)), contains three nested ‘for’ loops. These nested loops specify the data dependency of matrix elements. This data dependency causes difficulties in the parallelization of ILU because it brings forth communication overhead. In the diagonal form, a sparse matrix \( A \) is stored in vectors. In this work the penta-diagonal matrix is stored in the five vectors, viz., \( diag \), \( ldiag \), \( udiag \), \( lldiag \) and \( uudiag \). The matrices \( L \) and \( U \) are also stored in the similar way with prefixing letter ‘L’ and ‘U’ in the names of vectors. The outline of the newly developed algorithm for penta-diagonal ILU preconditioner is depicted in Fig.(2(c)). From this figure it is evident that in the new algorithm two nested loops have been eliminated. This elimination of loops optimize the algorithm as it reduce the computational cost of ILU. Furthermore, this optimize algorithm brings down the communication overhead and facilitates easier parallelization. A similar approach is applied to develop algorithm for the penta-diagonal version of ICH preconditioner.

Parallelization of K-S Solvers

As shown in Fig.(3), parallelization of K-S methods involves two main tasks, viz., data distribution and load balancing on different nodes of a parallel computer. In the case of iterative solvers the task of data distribution is to create a partition of the matrix. One of the features of this task is to divide vectors into parts, or broadcast.
the vectors to all nodes depending upon the type of the matrix partition. Generally, matrix partition is carried out in three different ways, viz., block partition, column-wise and row-wise partition. The classical way of load balancing is to distribute the loads equally among nodes of the parallel computer as indicated in Fig.(3) [10]. The flow chart for OOPP Solver is presented in the Fig.(4).

Data Distribution

There are three basic computational components of the K-S solvers, in particular, in the CGM solver. These components are: vector updates, inner products and matrix-vector products. All three components are parallelizable. A row-wise partition, as shown in Fig.(3), is used to decompose the matrix A. Therefore, the matrix is distributed across p processors as a block of rows. All vectors required for matrix-vector multiplication are replicated on all the processors. Each processor computes parts of matrix-vector product, or updates parts of the vector and passes it to the master. The master processor accommodates the parts of the vector in the whole vector at the appropriate location. A similar strategy is adopted for inner product operations.

Load Balancing

In this work, load balancing, as depicted in Fig.(3), has been implemented in a different way so as to facilitate the easier data distribution. Fig.(5) delineates newly devised load balance scheme. The new scheme incorporates a little extra load on the master node. As shown in Fig.(5), the extra load on master is n_1 where n_1 can be the either number of extra rows of a matrix or number of extra elements of a vector. The maximum value of n_1 is np – 1 where np is the total number of processors available. Therefore when matrix size is quite large, for example O(10^6), the value of n_1 is negligible. Using this new scheme master node divide the whole data by np, i.e. say dsize = n/np, and then sends the dsize amount of the data to slave processes. The remaining amount of data (dsize + n_1) is kept by master. And thus the new scheme reduces extra calculation and communication during data distribution.

Parallelization of Preconditioners

In the parallelization of the ILU and ICH preconditioners, nodes need data from others to which they are connected. As mentioned earlier, the data dependency is one of the main difficulties in the parallelization of ILU and ICH wherein the generated elements of the matrix are needed by another node. The present research work implements modified penta-diagonal ILU and ICH preconditioner algorithms. In these modified algorithms, two nested inner loops have been eliminated from the classical algorithms of ILU and ICH, which, in turn, reduces the overhead in regard to data dependency during parallelization.

Implementation of parallel algorithms, in this work, has been carried out using message-passing interface in C++ (MPI-C++) in a Single Program Multiple Data (SPMD) mode. In SPMD mode, all processors execute the same code on different data sets. A master-slave paradigm [3, 9, 10] for the parallel environment is adopted. Initialization of data (i.e. matrix A, vectors x and b) is imparted on the master processor. The initialized data is then distributed to all processors using an efficient load balancing scheme. All processors including the master execute the algorithm to their own part of data. If any need arises, slave processors also communicate to each other, for example in case of parallelization of ILU and ICH preconditioner. The master node also executes all other required I/O operations.

Results

A matrix is generated from discretization of 2D Laplacian equation and used as test matrix. The size of the matrix is n x n where n = 1002001. Fig.(6) depicts the comparison of number of iterations taken by the preconditioned and non-preconditioned CGM (Nonpre-CGM) solvers. The performances of preconditioners have been
measured by comparing the number of iterations. As evident from Fig.(6), the ILU preconditioner gives slightly faster convergence than that of the SSOR and DS preconditioners. In order to validate the results obtained from the parallel solver, residual errors from sequential and parallel implementations applied to test matrix are compared. This has been illustrated in Figs.(7, 8). The error in these figures is the norm of residual vector calculated at each iteration. It has been found that the results are in good agreement. The developed parallel penta-diagonal preconditioned CGM solvers are implemented on a Linux cluster. The cluster has 56 nodes and each of them contains 2 quad core processors of speed 2.66 GHz. Each node has 2 GB RAM.

One of the purposes of the experimentation is to record the computation time taken by solvers. Fig.(9) depicts the computation time taken by the Nonpre-CGM, DIAG-CGM, SSOR-CGM and ILU-CGM. It is observed from Fig.(9) that the computation time on 16 processors taken by the SSOR-CGM and ILU-CGM is less than that of the Nonpre-CGM and DIAG-CGM. Furthermore, the computation times taken by the SSOR-CGM and ILU-CGM are almost same whilst the number of iterations required by the ILU-CGM are less than that of the SSOR-CGM.

The speedup ($S_p$) factor based on the computation time is calculated using the formula $S_p = \frac{T_1}{T_n}$ where $T_1$ is the time taken by one processor and $T_n$ is the time taken by $n$ processors [6, 9]. The speedup factor obtained from developed solvers is depicted by Fig.(10). As indicated in Fig.(10) the ILU-CGM solver provides better speedup factor than that of other three solvers.

**DISCUSSIONS AND CONCLUSIONS**

In this paper, novel penta-diagonal versions of preconditioned and Nonpre-CGM solvers have been developed using the DS, ILU and SSOR preconditioners and applied for solving the linear system of equations obtained from the 2D Laplacian equation. Literature reveals that the ILU, ICH, SSOR and DS preconditioners are the most popular which are applied to the K-S solvers. Among these four, the penta-diagonal version of preconditioned CGM has been developed utilizing the ILU, SSOR and DS preconditioners. The chief aim of this work is to solve the system of equations obtained from the Laplacian equation by the penta-diagonal version of K-S solvers.
As the ICH preconditioner can only be applied to Symmetric Positive Definite (SPD) matrices with positive diagonal [4], the penta-diagonal version of the preconditioned CGM has been developed using the ILU, SSOR and DS preconditioners. It has been observed that the ILU preconditioner gives faster convergence as compared to the SSOR and DS preconditioners. However, the DS preconditioner gives poor convergence though it is easy to implement it.

Although the SSOR preconditioner takes few more iterations than that of the ILU preconditioner, the parallel implementation of the SSOR is simpler because of the absence of communication overhead. Furthermore, the computation times taken by both the ILU and SSOR preconditioners are almost the same. Thus, it is concluded that the computational cost of the ILU is more than that of the SSOR. Again, since the ILU-CGM results in better speedup as compared to the other preconditioned CGM solvers, the ILU is the most suitable preconditioner for the parallel pentadiagonal CGM solver. The new load balance scheme facilitates easier data distribution. In this scheme the master node holds little extra load but this extra load is negligible when the matrix size is quite large compared to number of nodes used for implementation.

Scope for Future Work

The scope for future work is directed towards testing of the developed solvers on large number of processors, generating the speedup data. The developed parallel pentadiagonal preconditioners can be applied to other popular K-S solvers. The results obtained from this work are useful for comparing the same obtained from the other OOPP K-S solvers applied to similar problems. Further, the classes developed may be reused for other code related to OOPP K-S solvers. These developed solvers should be implemented on bigger size matrices, e.g., n2 × n2 and n3 × n3 where n2 = 4004001 and n3 = 9006001. The bigger size matrices can be generated either by refining the grid during discretization of 2D Laplacian equation or using 3D Laplacian equation. Furthermore, these solvers can be applied to other multiphase flow problems such breaking waves or solitary waves [12, 14].

Acknowledgment

This work is partially supported by Microsoft(R) Ireland under the “Microsoft Postgraduate Research Scholarship 2008” program.

REFERENCES

SIMULATION IN EDUCATION
MODEL-BASED LEARNING IN ELECTRIC DRIVE DESIGN

Valery Vociovozov
Zoja Raud
Tallinn University of Technology
Ehitajate tee 5
19086 Tallinn, Estonia
zoja@cc.ttu.ee

KEYWORDS
Education, electrical engineering, database management system, object-oriented model.

ABSTRACT
The paper introduces an effective learning system to study the design process of electric drive. The package eDrive has been developed to demonstrate and support the major stages in electric drives design. This toolbox includes the database of motors, power converters, and gears, the object-oriented simulation software, the signal generation module, as well as the suitable visual interface. The system provides dc and ac drives development, tuning, and investigation with current, speed, and position adjustment in open loop and closed-loop modes.

INTRODUCTION
The range of problems soluble in the process of the electromechanical project development changes increasingly. Most of them did not enter the designing sphere some years ago, but now become the urgent tasks. In the first place, it concerns of design processes.

An electric drive is an electromechanical system that provides controlled inductive conversion of electrical energy into mechanical motion as Figure 1 shows. It consists of mechanical transmission $G$, electric motor $M$, electric power converter $C$, and informational controller $R$. The power of supply lines is transformed into motor supply energy by the electric power converter. Then, the motor converts it into electromagnetic energy, which in turn is converted into shaft mechanical energy. The controller forms the input references using information of the set-points, outputs, and disturbances (Vodovozov 2004).

The design goal is to build an effective system using the components proposed by many companies of world industry. Effective assembling of electric drive meets with problems owing to rather complicated algorithms (Roos et al. 2006, Cetinkunt 1991). A list of the design tasks includes equipment choice, structural synthesis, experiment planning and its result processing, as well as an implementation of designed systems. In the initial stage, the area of criteria is produced. Its space dimension is defined by a number of criteria. The set of electromechanical system parameters includes working forces, load torque, inertia, friction, speeds, accelerations, etc. Here, mechanical, electrical, electronic, and energetic problems are mixed.

To calculate and select equipment, the leading companies have developed their specific technologies. Examples are the guides and software of Siemens, Omron, Sew Eurodrive, Maxon Motors, Mitsubishi Electric, etc. Thus, a designer has an opportunity to choose a motor type, to impose restrictions, to estimate results, and to repeat an inquiry on other conditions (SEW Eurodrive 1998). Later, one can restructure a list of mechanical gears and power converters and conduct a search in appropriate databases to compose the optimum drive system. With the help of corporative databases, some combinations of parameters are optimized from the specific points of view and a scope of allowable environment condition is displayed to a certain data area. Then, the system tuning is executed (Hurrics 1994).

Such approach is usual for the majority of firms that carry out project designs and have rich experience in decision acceptance based on extensive computer databases, coming up to numerous catalogue archives and "absorbing" their contents and structures. These companies support educational processes in engineering schools and technical universities. Often, they dictate the teaching methods and approaches thus defining institutional curriculums. As a result, different engineering schools have different view on the design technology and instruments. The main drawback of such approach is the technological and data restriction that prevent building an optimal system. It is especially important on the first stage of the design process, when equipment is selected and the most responsible decisions are decided.

PROPOSED MODEL
A new universal model is discussed here that helps the student to understand the difference in companies’

![Figure 1: Diagram of electric drive](image-url)
approaches and to select the preferred company. The main features of an educational model that discern it from the models meant for the professional designers are:

- descriptiveness and compatibility of results;
- clearness of physical essence;
- suitability of report generation and format conversion;
- independence of particular company interests;
- matching the standards and design rules;
- availability of learning-oriented manuals, textbooks, and guides.

In a common sense, the actual state of an electric drive

\[ Y = F_0(X, U, E) \]  

(1)

is described by a particular function \( F_0 \). It takes into account the observed environmental conditions \( X \) and controls \( U \), as well as the non-observed disturbances \( E \). The domain of the function \( F_0 \) covers the set of variables \( \{x, u, e\} \) that depends on some energetic, power, and constructive restrictions. The required system behavior

\[ G = F_3(X, U, E) \]  

(2)

is described by a control algorithm \( F_3 \), that processes information \( I = (X, Y) \) about an environment and the load state \( Y \) and defines the control goal \( Z \).

A typical common used electric drive belongs to the inhomogeneous, non-capital, and unstable systems. Its object heterogeneity is reflected in the different nature of structural components and in various behaviors of informational and energetic processes. Any non-uniform structure may be represent by the decomposition of a system state vector using a limited set of linear-independent functions of inputs, such as

\[ Y = \sum (C_i F_i(X, U)) \]  

(3)

where: \( C_i \) is the \( i \)-th parameter of an object and \( F_i \) represents the referenced system of its input functions. In addition, one may describe a non-uniform dynamic environment by finitely-difference equations like this one:

\[ Y(t) = \sum (C_i Y(t - 1)) + \sum (C_j X(t - j + 1)) \]  

(4)

where \( C_i \) and \( C_j \) are the discrete factors, whereas \( X(t) \) and \( Y(t) \) involve the input and output values of the \( t \)-th discrete time interval. In such cases, heterogeneity deals with discrete nature of inputs and variables. The power pulse conversion on the base of continuous electromechanical transformation and continuous-discrete informational nature is referred to the same processes as well (Vodovozov and Loparev 2004).

Concerning the reaction on the different inputs, an electromechanical device is devoted to the nonlinear systems. In other words, its response to the sum of disturbances is not equal to the sum of responses to these disturbances:

\[ F_0(\sum (X_i, U_i, E_i)) \neq \sum (F_0(X_i, U_i, E_i)). \]  

(5)

Nonlinearity is caused by the limited controller and sensor code lengths, as well as by to the number of technical and program nonlinear chains and blocks. They lead to amplitude modulation of the digital-to-analog converters, distort the signal shapes, obstruct an adjustment of the power converters, saturate magnetic chains of electric machines, originate plugs, gaps, backslashes, and Coulomb or fluid friction in the mechanical gears and actuators.

Instability of electric drive deals with the time-dependence of the function \( F_0 \). It results in the parameter changing of informational system and in varying of the load torques and moments of inertia in mechanical systems. This feature serves as the background in the description of the object behavior as some time-dependent statements like these:

\[ U = U(t), X = X(t), E = E(t). \]  

(6)

Among the set of the states and behaviors of electric drive we should underline a class of program motions, which provides the achievement of the control goals assuming \( E = 0 \). The function that unites the desired system state \( Y \) with its actual inputs \( X, U \) is called here a system model:

\[ Y = F(X, U). \]  

In contrast to the uncertain function \( F_0 \), the model \( F \) has a particular state of the inputs \( X, U \) and the output \( Y \).

Describe the model by a pair

\[ F = (C, S) \]  

(7)

where \( C \) is a vector of model parameters and \( S \) is a structure that presents the composition and connection properties of its components. Then, the goal of an electric drive design may be formulated as an extreme problem

\[ Z(X, Y) \rightarrow \max. \]  

(8)

To solve this problem, it is required to find vectors \( C \) and \( S \), that ensure the maximum of a quality function \( Z \) available while the definite restrictions are furnished:

\[ C = \{C_{\min}, C_{\max}\}, S = \{S_{\min}, S_{\max}\}. \]  

(9)

Due to the nonlinearity, changeability, and heterogeneity of the vector \( Z \), the discussed optimal design problem is related to the class of multi-criteria problems of nonlinear programming.

Its solution is divided into three groups of tasks. First, it is a problem of possible (in the scope of the real context) minimization of a number of variants \( Q \). Then, it is required to find a structural and parametric system organization, which gives the optimal composition of the program control and the closed-loop control to adjust the system behavior. At last, the problem of informational and energetic system supply is solved. Using such approach to the matter, the project goal will be reached next to the defining the system structure and its constructive, energetic and informational features.

Due to the interconnection of the three listed groups of problems, the iterative approach is preferable here. On the
first stage, the set of restrictions $Q$ is proposed, a variant of basic structure is selected, and the parametric synthesis tasks are solved. Then, effectiveness of the solution is evaluated, and transition is being made to another structures. After that, the cycle is repeated with other limitations. Each such iteration includes

- definition and revision of the system quality index requirements;
- energetic synthesis of power equipment;
- structural design of a control system;
- dynamic synthesis of controllers and building the static characteristics;
- result evaluation and implementation discussion.

**SIMULATION**

The state-of-the-art of informational technologies opens the broad possibilities to build the models of electric drive. The most popular kinds of computer models are as follows:

- toolboxes that operate in common simulation environment;
- specialized software developed on the base of fundamental programming languages.

Among the instruments of the first group, such widespread tools are popular as MathLab and Simulink from MathSoft, MathCAD of MathWorks, ToolBook of Asymetries, LabVIEW of National Instruments, Electronics Workbench and Multisim of Interactive Image Technologies, Spice from OrCad, and Vissim of Visual Solution. Most of them are suitable for development and research the drive systems and their components. The main advantages of the listed systems are their powerful mathematical backgrounds, high quality of sheets, graphics, and computing data presentations, as well as an interconnection with hardware and operational systems with comfortable adaptation to computers of different styles and productivity.

In turn, the major benefits of the self-produced systems mastered using the standard programming languages are their exact orientation on specialists in electric drives. As a rule, they have moderate experience in modeling techniques and require the ready-to-use instruments suitable for engineering understanding. Compactness, high efficiency, and convenience are to be the significant features of such program modules. They are independent from additional simulation software and have no problematic interfaces with drive components, testing equipment, or real-time devices.

Contemporary development of nonlinear mathematical programming technology based on the computer power provides effective computation and optimization algorithms, different from those used in traditional simulation workshops. For this reason, we have selected the last approach to develop the project mastering system eDrive (Vodovozov and Loparev 2004). The described software offers the solution of the next typical educational and project management problems:

- informational support of selection electrical, mechanical, and electronic equipment;
- mathematical and computer simulation and full computation with database use;
- testing and result verifying in accordance with different criteria;
- the drive tuning and the load optimization.

**eDrive** includes next components:

- the powerful database;
- a set of adjustable controller schemes;
- the models of motors, converters, and gears;
- the graphic package for representation of steady-state and dynamic simulated processes with automatic and manual scaling, report generator, system analyzer, and preview facilities;
- the signal generator to produce the test reference and load signals as well as the nonlinear curves, noses, and filters.

**OBJECT-ORIENTED APPROACH**

A model is the main component of the developing, teaching, and maintenance procedures as well as the software core. From the educational point of view, it is of great importance to build generalized models having some redundancy to take account of future progress of the simulating object. That is why a uniform object description is considered here as the tool for design the generic classes (Vodovozov and Vodovozova 2002). Thanks to such approach, new modules will serve as the child classes that can be added to the system without changing the main model structure. The described software is the real example of the object-oriented technology (Usher 1996).

During the eDrive development, the integration of a large variety of models was carried out. We built a generic electric machine description as the base of an electric motor model of well-known and unknown types (Krishman 2001). In a conceptual idea, an ac motor, a brushless motor, and a dc motor are the particular cases of the selected model. From an object-oriented point of view, they are the descendants (children) of the base model.

By analogy, the generalized discrete converter description was developed as the base of the power converter model of known and virtual types. Transistor and thyristor converters of possible kinds are considered as the particular cases and the inheritors of the selected model. Such approach helps to develop the program system in accordance with the technical renovations.

As far as the mechanics simulation is concerned, there are many transactions devoted to the techniques of the body motion. In this field, our efforts have been directed towards the choice and linking the traditional models of the gears, the couplings, the hard and elastic transmissions with constant and alternating inertia, resistance, linear or nonlinear friction, etc.
The class library developed on the base of the C++ programming language comes up to ANSI standards and contains four basic classes shown in Figure 2:

- the electric motors class;
- the electronic converters class;
- the mechanical loads class;
- the result representations class.

The class Motor involves the private data members that store information in the calculating points, the model time steps, and the elastic deformation period of the mechanical part of the system. Moreover, it has the private functions that present the uniform two-phase electric motor model, which variables are the stator and rotor phase currents, the electromagnetic torques, and the rotation speeds of the rotor field and the loading shaft. The protected variables of the class Motor describe the type of power converter and load, the motor pole number, phase quantity, inductance and resistance, moment of inertia, electromechanical and electromagnetic factors. The list of protected member-functions includes the motor parameter functions that implement the procedure that solve the differential equations. Each model component may interact with the private and protected members via the public functions that send results into the file or onto the display screen.

The class DCEX has been inherited from the Motor. It serves as the model of a dc motor with an excitation circuit. Its members control the data that come in the computation process. Other inherited classes are DCPM (dc motor with permanent magnets), ACSC (induction motor with a square cage rotor), and VM (brushless servo motor). They have the same structure.

The next base class Trans describes the power supplies and electronic controllers of different kinds for the motors of the class Motor. The class Motor has been announced as a friend class for the class Trans, so it has an access into the private area of Trans. The class Trans encapsulates the private variables that describe the type of power supply as well as some private methods. They supervise the choice of an appropriate power source and describe the three-phase and dc supply mains with their inductance and resistance. Some polymorphic functions reflect the nature of the thyristor converters and the transistor modulators. The public variables of the class Trans collect information

![Figure 2: The library of classes](image)

![Figure 3: Database pane](image)

![Figure 4: Model window](image)

![Figure 5: Simulation result window](image)

![Figure 6: Signal generator window](image)
about frequency, voltage, resistance, and inductance of power converters and connect the class with other eDrive modules.

The base class Loads involves the models of different mechanisms. Its private area encloses such data members as the load type, load torque, elastic and tough factors, etc. The dispatch functions simulate the constant and the time-changing loads. The public area of this class includes the moment of inertia and gear description. The class Motor is announced as a friend of the class Loads.

The codes of all classes are closed to the end user. Nevertheless, they are the ready-to-use building blocks thanks to the suitable interface arrangement. Thus, programmers may use them to develop the new objects, as well as to improve and extend the models.

APPLICATION EXAMPLES

The described toolbox has been used to learn the development procedures of a number of drive systems having motor, gear, power converter, and controller for industrial, domestic, and medicine applications.

Figure 3 shows the first database window before the selection of motors, gears, and converters that match the requirements of the desired system. Sorting may be executed on any column. A short description of the selected equipment is presented in the separated area of the screen. The status bar keeps information about the number of equipment sets and the current group number. The Open button represents the selected equipment database where the particular machines, gears, or converters search is processed. The overall database size is as much as some thousands records with a dozen of text, numbers, and graphics data fields.

One of the model window pages is shown in Figure 4. The full information concerning the selected motor, load, source, and controller is presented here. A student may try different variants of parameters, speeds, loads, and controller tunings on these pages. He can save each result as a model file and open any previously saved file to continue the design process. The help system includes the detailed information about each mastering step.

Simulation accompanies the drive designing stages. The simulation results are placed in the three panes of the special window shown in Figure 5. One can evaluate a step response, static, and analytical figures, compare different variants, and select the best one.

The necessary input signals, nonlinear inputs, and load disturbances are arranged by mean of the signal generator that forms a dozen of signal shapes and distortions of various values and frequencies (Figure 6).

CONCLUSION

The educational problems concerning the synthesis of the complex drive system may be effectively solved using an object-oriented model in the learning process. The specific facilities of various engineering, technological and theoretical decisions integrated into the eDrive toolbox are offered in this paper. The technology improves the process understanding and leads to effective results. The equipment selection, simulation, and tuning methods affect both the structure and the dynamic performance of the developed drive. They have a significant impact on the educational capacity and the total cost of the desired system.

The paper introduces the uniform description as a universal tool of the class organization. Here, new modules may be implemented as child classes without any deconstruction of the main model architecture. The new ideas about the motor, converter, gear, and controller structure may be generated using these classes.

ACKNOWLEDGEMENT

This paper is supported by Estonian Science Foundation Grant G7572.

REFERENCES


AUTHOR BIOGRAPHIES

VALERY VODOVOZOV is an associated professor of Tallinn University of Technology and St.Petersburg Electrotechnical University, PhD, the author of 10 books, 25 inventions, and more than 100 papers and articles.

ZOJA RAUD is a PhD student and project supervisor of the Institute for Electrical Drive and Power Electronics in Tallinn University of Technology.
VISUALIZING THE EVOLUTION OF WAREHOUSES

Pietro Colombo
Alberto Coen-Porisini
Dipartimento di Informatica e Comunicazione, Università dell’Insubria
Via Mazzini 5, 21100 Varese - Italy
{pietro.colombo, alberto.coenporisini}@uninsubria.it

KEY WORDS
Visualization, Animation, Virtual reality, 3DML

ABSTRACT
Logistics applications for warehouse management take benefit from the integration of virtual environments that represent real warehouses. Virtual environments may facilitate the interaction of the user with the underlying logistics applications. Moreover, they ease analysis tasks on the distribution and the movements of the stored goods. This paper presents WarehouseEventViewer, an application that supports the analysis of the evolution of warehouses based on virtual environments that aim at facilitating and improving the effectiveness of the analysis.

INTRODUCTION
Industrial processes that manage the movement of the goods stocked in a warehouse are usually supported by complex logistics applications. Such applications, in turn, interact with information systems such as Enterprise Resource Planning (ERP) to deal with supply chain and warehouse management, to get detailed information on the stocked goods, their position, etc.

Some of the most advanced applications provide services both to allow a warehouse-keeper to get a detailed description of the path that he/she has to follow in order to reach the searched goods, and to keep trace of all the movements of the goods in a warehouse.

Virtual reality represents a complementary tool for such services, and is commonly used to facilitate their fruition. Interactive virtual warehouses show (with a certain approximation degree) the internal organization of real warehouses, and can be easily integrated into existing logistics applications (usually they are independent external modules). Moreover, they provide additional dedicated visualization and interaction capabilities.

During a time period, (e.g., a working day, a working week, etc.), the internal state of a warehouse continuously evolves. Once entered the warehouse, goods are registered, stocked, and moved by warehouse keepers according to the current state of the warehouse and storing policies.

Viewers integrated into logistics applications usually provide rich application programming interfaces (API) that allow the external applications to drive the visualization and the animation of a virtual scene. Taking advantage of such programming interfaces, viewers can be exploited in different ways: 1) they can be used for monitoring purposes, by keeping synchronized the state of the virtual scenes with the one of the real warehouse, 2) they can be used for simulation purposes, by visualizing hypothetical movements of the stored goods, or 3) they can be used to support warehouse management analysis tasks, by visualizing the evolution of a warehouse once one or more movements occurred. This work essentially focuses on the third usage mode. In such a scenario logistics applications record all the changes that occurred in a time period in the real warehouse, and generate commands to modify the characteristics of the visualized scene. Once all the commands are generated, they are sent to the integrated viewer that interprets them to show the evolution of the visualized warehouse by modifying the position and the visual characteristics of the elements of the scene. A viewer that supports such an interactive scenario, characterized by analysis and animation capabilities, is a fundamental tool for an analyst that needs to study what happened in a warehouse in a given time period.

The main goal of this work concerns the definition of an application that supports the visualization and analysis of warehouses evolution.

The proposed solution
In order to achieve such goal it is necessary: 1) to identify a modeling language suited for the warehouse domain, 2) to support the language by means of an interpretation engine, 3) to integrate the engine into a dedicated application that provides analysis services.

Scene modeling, which also includes the definition of behavioral aspects, must be as simple and quick as possible. Although several general purpose scene definition languages, such as X3D (Web3D Consortium, 2006) or its predecessor VRML (Web3D Consortium, 1997), could be used for defining virtual warehouses, their modeling approach is expensive and time consuming. On the contrary, in our approach scene modeling exploits the simplicity and the modularity of warehouses in order to minimize the efforts.

We propose an approach based on 3DML (Flatland Online Inc, 2003a), a 3D scene definition language (see Section 2), which, although not widely known, is well suited for our goals. Scenes are defined in two steps: first, and once-and-for-all, the used subparts and their details are created; in a separate phase, these parts are assembled into the global scene. The concept is that only the latter part is needed to construct any new instance (but the first one) of a scene in a given application scenario. The language besides providing modeling constructs for structuring scenes, provides constructs to manage dynamic changes, and basic interaction with the users that navigate the scenes.

The 3DML modeling approach is supported by Open3dml (Colombo et al. 2007) (see Section 3), our open-source interpretation engine that supports the visualization and navigation of 3DML scenes. Open3dml was originally designed as an ad hoc viewer for web based logistics applications, then the engine was extended with a rich API and advanced communication capabilities.

Finally, the solution is completed with WarehouseEventViewer (WEV) (see Section 4), an
application that we specifically designed to support the analysis of the evolution of warehouses. WEV is built around Open3dml, which acts as interpretation engine for 3DML warehouse scenes. WEV provides different tools to study the evolution of warehouses and to visualize their animation.

The paper is organized as follows: Section 2 introduces the modeling approach for warehouse scenes and shortly describes 3DML; Section 3 presents the characteristics of Open3dml, an interpretation engine for 3DML scene; Section 4 describes WarehouseEventViewer, an application built around Open3dml that supports the analysis and visualization of the evolution of warehouses; finally Section 5 draws some conclusions and discusses the future work.

**3DML**

The 3DML language is an XML (Bray et al., 2006) based language specifically designed to describe 3D virtual environments in a direct and easy way. It has been defined by Flatland Online and focuses on 3D content creation for web applications.

3DML provides different kinds of constructs that allow a modeler to define both the static and the dynamic aspects of a virtual environment. The characteristics of 3DML makes the language suited for different application scenarios (Colombo et al. 2007; Coen-Porisini et al. 2007; Polovina et al., 2000).

A 3DML scene, referred to as a spot by the 3DML community, is a 3D structure composed of instances of blocks disposed over a map. A map is a regular 3D grid of textual labels that indexes a set of blocks. A block is a 3D polygonized structure with textures and/or colors. Each block is defined by extending an abstract counterpart taken from blocksets, which are external repositories of 3d elements related to a modeling domain.

An abstract block defines the basic structural characteristics of the element that will be used in a spot, that is it can be thought as the skeleton of the block. It is composed of a set of “sub-parts”, each of which is geometrically structured by polygons and optionally provides appearance attributes (e.g., textures and/or colors).

Abstract blocks cannot be directly inserted into a scene. An abstract block must be extended into a concrete block in order to be visualized. An extended block defines a new identifier, maintains the geometrical characteristics of its abstract counterpart and possibly redefines some of the appearance attributes associated with its components. As an example, a concrete block may have the same structural characteristics of its parent, while differing for the orientation of an internal part.

Figure 1 shows a block representing a shelf.

Moreover, extensions can define transformation such as rotations, translations and scaling on the single components of a block.

Blocks can define listeners for particular types of navigation or temporal events (e.g., mouse click, mouse over). Such listeners can trigger the execution of the transformation for the instances of the blocks on which they are defined. Once blocks, are defined, the scene is designed by defining its internal map. The map is built by instantiating the extended blocks and positioning the resulting instances in a specific location of the 3d grid. The grid is defined level by level by specifying for each row and for each column whether the corresponding cell is empty or is an instance of an extended block.

**Scripting in 3DML**

3DML allows the modeler to embed scripts for defining functions to access the elements of a scene. The supported language for such activity is a Simkin dialect.

Simkin (Whiteside, 2000) is an embeddable scripting language whose scripts can be inserted in any XML document, including 3DML ones. A Simkin extension called Roverscript (Flatland Online Inc, 2003b) is specifically designed for the 3DML language.

Roverscript defines several elements and services to access the properties of the elements in a spot. Among those, the player provides methods for changing the position of the avatar in the virtual scene, the map gives access to the instances of blocks in a given position, and the block provides services to modify the properties of block instances by means of transformations such as translations, rotations, scaling and overriding of appearance attributes.

This allows one to define, inside a 3DML document, functions that can be invoked during the scene navigation. More specifically they can be triggered by 3DML navigation events, by GUI events, by timers.

**OPEN3DML**

Open3dml is an open-source application designed to support the visualization and navigation of 3D environments described in 3DML.

Open3dml is based on Java3D, a well established development technology for 3D graphics that 1) achieves good rendering performances via graphical hardware accelerations, 2) ensures high portability, hiding platform dependent implementation aspects, and finally 3) provides an unified application interface for the underlying OpenGL or DirectX APIs. As a result, it is platform independent, needs a Java plug-in and a Java3D package installed, and features real time GPU accelerated renderings.

Open3dml is composed of three modules that support scene loading, scene rendering and user interaction management, respectively.

**Scene loading**

Scene loading comprises the following phases:

1. Parsing of the 3DML scene (an XML document) and definition of a DOM tree where each 3DML element directly corresponds to a node of the DOM. The
generated structure provides services to access the single nodes of the structure.

2. Enriching the DOM tree by generating dedicated structure around the DOM nodes. Each node is extended in order to provide services to access the characteristics of the correspondent 3DML element.

3. Definition of the Java 3D objects. Open3dml analyzes which blocks will be used in the final scene, and generates their structure by instantiating Java3d objects. Such objects are preprocessed to optimize their subsequent rendering (see next paragraph).

4. Scene composition. According to the structural characteristics of the 3DML spot, the Java3d objects defined in the previous step are cloned and positioned into the final scene.

Navigation and Rendering
The Open3dml real-time rendering engine takes advantage of the high level structure of the 3DML scene being displayed. Each defined block type can be pre-processed and stored in a display-list or a vertex-buffer-object list, to be efficiently referenced multiple times in the same scene (through Java3D Link-ShaderGroup mechanism).
To improve visualization effectiveness, our scene can include semi-transparent (alpha blended) elements: luckily it is easy to render the blocks composing the scene in a depth sorted way thanks to the 3D lattice structure of block references.
Visibility culling and view frustum culling are also carried out in a straightforward way, as the 3D lattice constitutes a natural decomposition of the scene in cells for pre-computed visibility.
Collision detection is implemented in an easy and efficient way: at scene loading time each block is categorized as "impassable", "floor tile", "empty space", and so on; this information is accessed at navigation time, avoiding any further computational effort. This is clearly an approximation, but it is well suited for our purposes. Collision detection is used both for standard collision responses, and as a possible trigger of scripted events (e.g., "step in" trigger).
The scene can also integrate animated textures (described in the 3DML document as GIF images): all frames are preloaded in the graphic card memory to be later alternated.

Functions and triggers
Open3dml integrates a SimKin interpreter (Whiteside, 2000), an open-source interpreter for RoverScript scripts that are embedded in a 3DML spot. This module is composed of: (1) a parser that analyzes the Roverscript code; (2) a set of Java classes to represent 3DML elements; and (3) an engine that executes the parsed code by accessing the properties of the instances of the above mentioned classes.
RoverScript functions allow one to modify the position of the blocks that compose a virtual scene. As an example consider the following code in which a Roverscript function named transport_block operates translations and rotations on the blocks of a spot.

```
<string value=Open3dml.processScript("transport_block(10,1,3,12,4,8,112,0,111,45,3)"")>
```

which describes the invocation of the method processScript in order to execute the previously defined transport_block Roverscript function. As a consequence, the current scene is modified by moving and rotating the block originally positioned in 10, 1, 3 of the 3d map.
The interpretation engine parses the caller request before executing it. If an error occurs, the execution of the called function is aborted and a detailed error message is returned.

WAREHOUSE EVENT VIEWER
WarehouseEventViewer (WEV) is an open-source Java application that supports both the analysis and the animation of the internal evolution of a warehouse. WEV integrates Open3dml as a visualization and navigation module for virtual warehouses defined in 3DML.
Logistics applications for warehouses management are usually based on a model representing the real warehouse. In other words, such model describes the state of the warehouse, hence any event occurring in the warehouse causes the model to be updated. The integration of 3D visualization capabilities into the logistics application requires the introduction of an additional model representing all the visual aspects. Therefore, it will be necessary to keep updated both models with the current state of the real warehouse.
In our approach visual models are 3DML scenes that reflect both the structural characteristics of the real warehouse and the position of the stored goods. Open3dml allows external applications to affect the visualization by modifying the characteristics of the scene at visualization time. As a consequence, the evolution of a virtual warehouse can be completely controlled by an external application that invokes RoverScript functions to affect the visual model.
WEV supports two different interaction modes named online and offline, respectively. In the former mode the logistics application invokes the RoverScript functions each time an
event occurred in the real warehouse causes the local model to change. In this way both the local and visual models are kept consistent. In the latter mode, the logistics application stores all the occurred events in order to provide WEV with the information needed to visualize the warehouse evolution later on. In this way the user can review what happened in the warehouse and can possibly perform any kind of analysis. Notice that while the former interaction mode was already supported by Open3dml, the latter has been implemented in WEV.

In order to operate offline, the information that needs to be stored concerns the animation events that cause the visual evolution of the warehouse scene. An animation event is characterized by a Roverscript function that modifies the visual model, along with two time stamps representing when the function has to be invoked and when the function terminates.

**Animation events**

Animation events are represented in XML and stored into textual files that contain all the input information for analyzing the evolution of the warehouse with WEV. WEV manages animation events files that are compliant with the following XML document structure.

The root element of the document is named `warehouse_log`. Such element defines the attributes `id` and `scene_url`, `id` stores a value that is used as an identifier for the whole sequence of events. `scene_url` specifies the URL of the 3DML scene representing the visual model of the warehouse. `warehouse_log` contains a sequence of elements of type `transport_block`, which in turn specifies the single animation events. `warehouse_log` comprises the following attributes:

- `x_source, y_source, z_source` to specify the position of the block that is the target of the animation event. These coordinates are expressed in term of indexes of the 3d map associated with the spot
- `x_dest, y_dest, z_dest` to specify the final destination of the movement, i.e., the cell of the 3d map where the block has to be positioned
- `x_off, y_off, z_off` to specify the offset expressed in terms of distance from the middle point of the 3d map cell of the final destination point
- `angle_a, angle_b` to specify rotation angles (expressed in degrees) respectively respect to the x and y axes.
- `enabled_at` to specify the timestamp associated with the starting of the animation, that is when the animation can be invoked
- `duration` to specify the duration of the animation
- `tracement` to specify whether at visualization time the animation simply shows the original and the final position of the animated block or all the intermediate positions.

**WarehouseEventViewer GUI**

WEV provides a user-friendly GUI that integrates tools that can be used to visualize 3DML spots, to navigate them and to manage the evolution of the scene. WEV GUI is implemented using Java SWING technology. Figure 2 shows a snapshot of the WEV graphical user interface. WEV GUI, is designed to support the study of the evolution of warehouses. The control and visualization services associated with the animation of the evolution of warehouses represent the most advanced functionalities provided by the application. Such functionalities allow one to show the whole evolution of a warehouse during a certain time period. WEV allows an analyst to load textual files compliant with the previously introduced format. WEV parses the XML document and generates an internal model used to coordinate the animation. Such model keeps track of all the animation events that characterize the evolution of the warehouse. Every animation event is characterized by two timestamps. The first one specifies the time instant when the animation command starts being processed, while the second one when such processing ends. Notice that the whole animation service is built around a synchronization mechanism that exploits the temporal information associated with the animation events.

The internal model associated with an animation is a structure composed of 1) an attribute that stores the current time value, 2) a 3DML spot that represents the warehouse and 3) an ordered set of nodes, where each node refers to a time stamp and to a list of scripts that are enabled from the time instant specified by the time stamp. The stored scripts contain the definition of the invocation of Roverscript functions that are defined in the 3DML spot.

Open3dml supports the visualization and navigation of the previously mentioned 3DML spots. The visual frame operated by Open3dml is integrated into the GUI by means of a dedicated internal frame.

Starting from the internal model, the whole system that controls the animations is defined. The layout of the interface is similar to the one of a common multimedia player. More specifically, the core element of the interface is a slide bar that manages the evolution of the animation. Such tool operates the time model associated with the animation, which in turn, is associated with a queue of animation events. By operating the cursor of the slide bar it is possible to modify the time value that is referred to by the internal model. Moreover the position of the cursor is continuously synchronized with such value by means of listeners. The time instant referred to by the cursor of the slide bar does not necessarily correspond to one of the timestamp associated with the animation events. All the time instants that are contained in the range specified by the start and the final instant associated with an animation event, can be reached by operating the slide bar.

The value associated with the initial position of the slide bar specifies the lowest value associated with the animation events. Conversely the final position of the slide bar refers to the maximum value associated with the time stamps of the animation events.

The animation can be controlled also without directly manipulating the slide bar by using the buttons PLAY, PAUSE, STOP, FW, RW, NEXT, PREV.

The behavior associated with such buttons is quite intuitive:

1. **PLAY** activates the animation starting from the time instant that is currently referred by the internal model and by the cursor of the slide bar. The activation operates the time value of the internal model. Accordingly the cursor of the slide bar changes its current position.
2. **PAUSE** stops the animation by freezing the time value of the internal model. As a consequence the movement of the cursor is blocked.
3. **STOP** stops the animation, modifies the current time values of the internal model by setting it to the lower timestamp associated with the animation events. As a consequence the cursor of the slide bar is moved to the initial position.

4. **FW** and **RW** increment and decrement the current time value of the internal model, respectively. Consequently the position of the cursor is updated.

5. **NEXT** and **PREV** operates on the time value of the internal model. **NEXT** (**PREV**) allows the user to set the time to the lowest (greatest) timestamp among those that are greater (lower) than the current time stamp value associated with the nodes of the internal model.

A further tool, named Configuration, allows one to configure the behavior associated with the buttons **PLAY**, **FW**, and **RW**. It provides two slide bars respectively named **Speed** and **Time granularity** that can be used to modify the animation fluency. **Speed** allows one to set the animation speed by modifying the time interval between two consecutives updates. This setting directly operates the behavior associated with the button **PLAY**, and consequently the speed of the cursor of the animation slide bar.

**Time granularity** allows one to configure the time length of a basic animation step, that is the value that is added at each step to the time value of the internal model at animation time. This setting affects the actions associated with the buttons **PLAY**, **FW** and **RW** and as a consequence the behavior of the animation slide bar.

A detailed list of all the timestamps associated with the animation events is shown in a dedicated frame named **Timestamp list**. The represented timestamps correspond to the value stored by the nodes of the internal model (i.e., the time instants when the animation commands start being processed). Timestamps are ordered and visualized in dedicated cells. Each cell provides a textual label that reports the timestamp value, and an additional label that provides the number of animation events enabled at the time instant specified by the timestamp. Notice that such value reflects the number of animation events in the corresponding list of the internal model. Cells can be selected by clicking on them. Such selection modifies the time value of the internal model, and the position of the cursor of the animation slide bar is updated. Moreover, a mechanism based on the usage of listeners synchronizes the current selected item with the time value. At animation time the cell characterized by the greatest timestamp whose value is less or equal to the time value of the internal model is highlighted.

The selection of a cell triggers both the execution and the visualization of the associated animation events by means of an additional frame named **Events at Timestamp**. Such frame is organized as a list of cells labeled with a short description of the animation event. A more detailed description of such actions are shown in dedicated dialog boxes whenever one of the corresponding cells is selected.

The visualized information is the following:

1. **ID**, a univocal identifier
2. **Timestamp**, an attribute that specifies the timestamp associated with the animation event.
3. **Type**, an attribute that specifies the event type, that is whether the event is the start or the final event for the whole animation, or if it corresponds to an animation command for an element of the visualized scene.
4. **Content**, an attribute that provides the code to invoke a Roverscript function defined in the visualized spot.

Notice that the effect of the animation strictly depends on the operations performed by such functions.
Animation
The animations are activated whenever the time value of the internal model is in the range specified by the timestamps associated with an animation event. The activation triggers the invocation of a Roverscript function. Notice that Roverscript functions modifies the positions of the blocks in the spot, and their processing is atomic and independent from any duration value. As a consequence, without applying any other mechanism whenever a transport function is invoked, the involved block is instantaneously transported from the initial position to the final one. In order to overcome such weakness an additional mechanism was introduced. For each movement associated with an animation event, a temporal information is added to the initial and to the final positions, hence, intermediate positions are calculated by means of linear interpolation techniques.

Configuring the visualization
Open3dml hides popping artifacts by means of a shadow line that hides the bounds of the visualization. Notice that the distance parameter do not affect the rendering performance, but if correctly set, it allows one to improve the realism of the visualization.

Open3dml also supports the configuration of the view depth. A proper configuration allows one to improve the performance of the rendering. In case of a platform with low computational capabilities, good rendering performances can be guaranteed by decreasing the value of this parameter.

WEV supports configuration of both the view depth and the shadow line by means of a dedicated menu of the Configuration frame. The control is operated by two slide bars that modify the current value of the distance of the shadow line, and the current view depth respectively. Such tools directly operate the corresponding parameters used by Open3dml.

CONCLUSIONS
This paper introduced WarehouseEventViewer (WEV), an application that supports the visualization and analysis of the evolution of warehouses.

WEV provides tools to visualize and to navigate virtual warehouses defined by means of an approach based on the 3DML modeling language.

WEV integrates Open3dml, an interpretation engine for 3DML scene, and several tools that allow a user to analyze the single events that occurred during the evolution of a warehouse. WEV also supports the animation of such evolution.

At present WEV is in a prototypical state, several refinements are currently under development. Future work concerns the release of the project to the open-source community, and the usage of the application in an industrial context. Moreover, we plan to experiment the usage of WEV in application contexts different from logistics.

REFERENCES


ANALYTICAL AND NUMERICAL SIMULATION
ANALYSIS OF WAITING TIMES IN CELLULAR MOBILE SYSTEMS WITH RETRIALS AND GUARD CHANNELS

M.J. Lopez Herrero
School of Statistics
Complutense University of Madrid
28040 Madrid,
Spain
e-mail: lherrero@estad.ucm.es

KEYWORDS
Cellular mobile networks, Queueing, Retrials, Guard channels, Waiting times, First-passage analysis

ABSTRACT
A large number of studies of modern cellular mobile networks concern performance measures directly computable from the stationary state probabilities, such as the blocking probability, the mean traffic rates of the various kinds of calls, etc. In this paper we consider a cellular mobile system with retrials and guard channels for the handover calls, but we concentrate on the waiting times of the customers. More concretely, we first build a Markovian model representing a station of the network and then we study the waiting time of a call. This measures shed light in the behavior of the system and quantify the quality of service from the point of view of the customer.

INTRODUCTION
In the studies of cellular mobile communication networks, it has been recognized that the proper modeling of customer behavior is very important for the reliable evaluation of the quality of service (QoS) of a given system. Thus, fresh calls, initiated inside a given cell are distinguished from handover calls that appear due to the mobility of the customers. Since handover calls already use network resources, they should be prioritized with respect to fresh calls. Indeed, a premature termination of an ongoing call is considered to degrade the QoS much more severely than the experience of a busy signal at an attempt to initiate a call. To cope with this problem, several approaches have been proposed in the literature. One efficient method is the use of guard channels which are reserved only for the handover calls. The performance evaluation of systems with guard channels goes back at least to the pioneering paper of Guérin (Guérin 1988). Subsequently, many authors have studied the performance of more involved systems with guard channels.

To represent accurately a cellular station, the phenomenon of repeated attempts should be also taken into account. Indeed, due to the increasing number of users and the system complexity of modern cellular networks, the impact of retrials is no longer negligible. For that reason, several papers consider 2-dimensional models to represent the behavior of such systems (see e.g. Tran-Gia and Mandjes 1997; Ajmone Marsan et al. 2001; Liu and Popojuwo 2006). Indeed for the accurate representation of a system with retrials we need at least two state variables, one recording the number of busy channels and the other recording the number of calls in the retrial orbit. Although the computational complexity of these 2-dimensional models is much greater than that of approximate 1-dimensional models resulting from state-aggregation, the corresponding results are far more reliable.

The 2-dimensional continuous time Markov chains that appear in systems with retrials are in general harder to analyze than their standard counterparts (see e.g. Artalejo and Falin 2002; Falin and Templeton 1997; Artalejo and Gómez-Coral 2008). This happens due to the retrial feature, that implies linear transition rates with respect to the number of customers in orbit. Because of these mathematical difficulties most studies for systems with retrials concern performance measures directly computable from the stationary probabilities of the number of customers in the system. Such measures include the blocking probability, the mean traffic rates of the various kinds of calls, etc. In particular this is the rule in the recent studies for cellular mobile networks with guard channels and retrials.

In the following pages we describe the model for a single cellular station and we develop the algorithmic schemes for the waiting time analysis. We first obtain a linear system for the Laplace-Stieltjes transforms of the conditional waiting times and then proceed to the unconditional waiting time. We also study the moments of the waiting time. Finally, we comment on the results and discuss possible extensions.
THE MODEL

We consider a cellular mobile network and we concentrate on a single station. The arrival processes for the fresh and the handover calls are assumed to be independent Poisson processes with rates $\lambda_F$ and $\lambda_H$ respectively. There exist $N$ channels for serving the calls. A number $G$ of them are guard channels reserved only for handover calls, while the rest, $N-G$, are regular channels that can serve both fresh and handover calls. A handover call that finds all channels busy departs from the station and it is lost for ever. A fresh call that finds at least $N-G$ occupied channels, joins the retrial orbit with probability $\theta$ or it is lost with probability $1-\theta$. The calls in the orbit conduct retrials. The interretrial times for the calls in the orbit are independent exponentially distributed random variables with parameter $\alpha$. If a retrial call finds at least $N-G$ occupied channels, it returns to the orbit with probability $\theta$ or abandons the system with probability $1-\theta$. The completion rate for each call in service is $\mu$. We denote by $\lambda = \lambda_F + \lambda_H$ the total arrival rate of non-retrial calls.

The state space to $S = \{0, 1, \ldots, N\} \times \{0, 1, \ldots, K\}$. The orbit capacity $K$ is chosen large enough to guarantee the numerical convergence of the mean values. An illustration of its transition diagram is presented in Figure 1, for the special case $N = 6$ and $G = 3$.

The transition rate matrix of $\{(C(t), N(t))\}$ is block tridiagonal and the computation of the stationary probabilities can be done at a low cost using one of the standard algorithms. Moreover, the blocks of the process are tridiagonal matrices so we can also use the algorithms of Servi (Servi 2002) which exploit further this special structure. We do not proceed further in the stationary analysis of the system, as our focus is on the waiting time that we present in the next section.

THE WAITING TIME OF A CALL

We consider the system at an arbitrary time $t$ and suppose that it is at state $(i, j)$, $j > 0$. We mark one of the $j$ retrial calls and denote by $W_{i,j}$ a random variable representing its waiting time in the system (excluding service).

![State Space Diagram](image)

Figure 1. State Space and Transitions for $N=6$ and $G=3$

Let $C(t)$ and $N(t)$ be respectively the number of busy channels and the number of calls in orbit at time $t$. Then $\{(C(t), N(t))\}$ is a continuous time Markov chain with infinite state space $\{0, 1, \ldots, N\} \times \{0, 1, \ldots\}$. To numerically solve the system of the balance equations and the linear systems for the Laplace-Stieltjes transforms of several first-passage times of the model, we truncate the state space to $S_a = S \cup \{a\}$, where $a$ represents an absorbing state. The absorption occurs when the waiting time of the tagged customer expires. The dynamics of this auxiliary absorbing
Markov chain where the absorption corresponds to the termination of a waiting time are illustrated in the transition diagram given in Figure 2, where \( \delta_{x,y} \) is the Kronecker’s delta which is 1 for \( x = y \) and 0 otherwise.

Being on a state \((i, j)\) we now condition on the time to the next event and the next state, that is we apply first-step analysis in the auxiliary chain to obtain the Laplace-Stieltjes transforms \( W_{i,j}^*(s) \). Then, the transforms \( \{W_{i,j}^*(s)\} \) are given as the smallest non-negative solution of the following linear system of equations having a block tridiagonal coefficient matrix of the unknowns.

\[
\begin{align*}
(s + \lambda + i\mu + j\alpha)W_{i,j}^*(s) - \lambda W_{i+1,j}^*(s) \\
-\lambda W_{i,j-1}^*(s) - (j - 1)\alpha W_{i,j-1}^*(s) = \alpha,
\end{align*}
\]

\[0 \leq i \leq N - G - 1, \quad 1 \leq j \leq K,
\] (1)

\[
\begin{align*}
(s + \lambda_H + (1 - \delta_{j,K})\theta\lambda_F + i\mu + j(1 - \theta)\alpha)W_{i,j}^*(s) \\
-\lambda_H W_{i+1,j}^*(s) - (1 - \delta_{j,K})\theta\lambda_F W_{i,j+1}^*(s) - i\mu W_{i,j-1}^*(s) \\
-(j - 1)(1 - \theta)\alpha W_{i,j-1}^*(s) = (1 - \theta)\alpha,
\end{align*}
\]

\[N - G \leq i \leq N - 1, \quad 1 \leq j \leq K,
\] (2)

\[
\begin{align*}
(s + (1 - \delta_{j,K})\theta\lambda_F + N\mu + j(1 - \theta)\alpha)W_{N,j}^*(s) \\
-(1 - \delta_{j,K})\theta\lambda_F W_{N,j+1}^*(s) - N\mu W_{N,j-1}^*(s) \\
-(j - 1)(1 - \theta)\alpha W_{N,j-1}^*(s) = (1 - \theta)\alpha,
\end{align*}
\]

\[1 \leq j \leq K.
\] (3)

We can also exploit this system to develop a recursive algorithm for computing the moments \( E[W_{i,j}^n] \) of the conditional waiting times up to any desired order \( n \).

\[
W_{i,j}^*(s) = 1 - \theta \sum_{i=N-G}^{N-K} \sum_{j=0}^{K-1} p_{i,j} + \theta \sum_{i=N-G}^{N-K} \sum_{j=0}^{K-1} p_{i,j} W_{i,j+1}^*(s).
\] (4)

Thus, once the system for \( W_{i,j}^*(s) \) has been solved, we can compute \( W_{i,j}^*(s) \). Moreover, for the corresponding moments we can obtain

\[
E[W^n] = \delta_{n,0} + \frac{n}{\lambda_F} \sum_{i=0}^{n} \sum_{i=1}^{K} \lambda_{i,j} E[W_{i,j}^{n-1}].
\] (5)

that is, the \( n \)-th moment of the unconditional waiting time is expressed in terms of conditional moments of smaller orders. In particular, for \( n = 1 \) we obtain \( E[W] = \frac{1}{\lambda_F} \sum_{i=0}^{N} \sum_{j=1}^{K} \lambda_{i,j} \), which can be independently derived by applying Little’s formula for the number of calls in orbit. For \( n = 2 \) we obtain the formula \( E[W^2] = \frac{2}{\lambda_F} \sum_{i=0}^{N} \sum_{j=1}^{K} \lambda_{i,j} E[W_{i,j}] \) which expresses the second moment of the unconditional waiting time in terms of the mean values of the conditional waiting times.

\[
\begin{align*}
(i, j + 1) \\
(i + 1, j) \\
\lambda \\
\alpha \\
(j - 1)\alpha \\
(i + 1, j - 1)
\end{align*}
\]

\[
\begin{align*}
(i - 1, j) \\
\frac{i\mu}{\alpha} \\
\lambda \\
(i + 1, j)
\end{align*}
\]

\[a) \ 0 \leq i \leq N - G - 1, \quad 1 \leq j \leq K
\]

\[
\begin{align*}
(i - 1, j) \\
\frac{i\mu}{\alpha} \\
\lambda \\
(i + 1, j)
\end{align*}
\]

\[
\begin{align*}
(i + 1, j) \\
\frac{i\mu}{\alpha} \\
\lambda \\
(j - 1)\alpha \\
(i + 1, j - 1)
\end{align*}
\]

\[b) \ N - G \leq i \leq N, \quad 1 \leq j \leq K
\]

Figure 2. Transitions out of \((i, j) \in S^\alpha\)

CONCLUSIONS AND EXTENSIONS

We have complemented earlier studies of cellular mobile networks with retrials and guard channels by providing an approach for the computation of waiting times. In combination with the classical performance measures in terms of the stationary state probabilities, e.g., blocking.
probabilities, mean traffic rates etc.), we can give a better quantification of the QoS of a cellular mobile network. The approach is based on the definition of certain auxiliary absorbing Markov chains, where the absorption corresponds to the termination of a waiting time. Then, using first-step analysis, we obtain block tridiagonal linear systems that can be solved easily with well-known algorithms. The approach has been used to compute some first-passage time performance measures: time to a guard channel activation and the time till the next lost call. Interested readers can found a detailed description and numerical results in the paper Economou and Lopez-Herrero 2008. This approach can be also applied in cellular mobile networks with more involved call handling schemes (see e.g. Basharin and Merkulov 2003 and Liu and Fapojuwo 2006). For example, it is possible to study a model that takes into account the portion of the blocked handover calls that retry later for establishing a connection. The dynamics of the system becomes much more involved as we have a 3-dimensional state-description. Nevertheless, the corresponding random variables can be studied similarly by defining the associated auxiliary absorbing Markov chains and performing first-step analysis.

Acknowledgements. The support of the research project MTM2005-01248 is gratefully acknowledged.

REFERENCES


AUTHOR BIOGRAPHY

M.J. LOPEZ-HERRERO receives her Ph.D. degree from the Autonoma University of Madrid, Spain in 1994. Since 1987 she has been faculty member in the School of Statistics at the University Complutense of Madrid. She is an active member of the Queueing Modelling Group. She works in the fields of Queueing Theory and Telecommunication Systems with an emphasis on retrial queues, matrix-analytic and maximum entropy methods. Web address http://www.mat.ucm.es/~mcqt/qmg/lopezh.html
A Bandwidth Management Framework for Wireless Mobile Ad Hoc Networks

Benny M Njambo
Gerrit K. Janssens
Transportation Research Institute (IMOB)
and
Wim Lamotte
Expertise Centre for Digital Multimedia (EDM)
Hasselt University, Diepenbeek, Belgium
Email: benny.njambo@student.uhasselt.be, {gerrit.janssens,wim.lamotte}@uhasselt.be

KEYWORDS
Mobile ad hoc networks, Quality of Service, Bandwidth Management

ABSTRACT
Quality of Service (QoS) in mobile ad hoc networks is a very important issue, especially these days as multimedia traffic finds its way on the network. Bandwidth is a very scarce commodity on wireless networks so bandwidth management is a very important component of QoS in ad hoc networks. In this paper a bandwidth management framework for mobile ad hoc networks is presented. Traffic is grouped into classes according to their priority. When a flow is using more bandwidth than its required minimum, it can release some bandwidth to a new flow which requires the bandwidth. So when a new flow requests for admission, the framework should check the bandwidth allocation of the on-going flows. When the available bandwidth is less than the required minimum of the flow, then bandwidth has to be released by the on-going flows starting with the least priority flow. When the highest priority flow releases its extra bandwidth and the bandwidth remains inadequate then the system goes into panic mode, in which flows are paused, starting again with the least priority flow.

INTRODUCTION
Mobile Ad Hoc Networks represent future generation wireless networks, with a high degree of versatility and robustness, capable of being deployed quickly and economically at places lacking any infrastructure. The characteristics of these networks such as bandwidth scarcity and fluctuations, node mobility, hostile working conditions and battery power constraints, have hindered their development. Projected applications of such networks include defense-based applications (war scenarios), disaster relief operations (earthquake, rural areas), commercial applications (home networking, extending Internet connectivity, inter-vehicular networks) and sensor networks [3]. Many of these applications require a distinction in the quality of connections being supported in terms of bandwidth availability, end-to-end delay and jitter [1,2]. As multimedia traffic find its way into wireless networks, the use of UDP transport layer protocol alone does not suffice to the needs of users. As the number of UDP supported traffic increase the throughput of each flow is drastically reduced as shown in Figure 1. In this case three flows were simulated under ns-2 simulator and plotted with x-graph. Most multimedia traffic requires a more stable throughput for them to be regarded to be useful [8]. Therefore there is a dire need for a QoS model integrated within the nodes for such applications. The model must be able to distinguish flows based on their QoS needs and have mechanisms that work to meet those requirements. Since there is no central coordinator, the QoS model must operate in a fully distributed manner. Such requirements combined with the hostile working conditions of an ad hoc network make the task of designing such a model very challenging [9]. We propose a novel QoS model that differentiates the flows into classes and attempts to provide bandwidth and delay guarantees to flows of highest priority class.

Figure 1 Graph showing the throughput for three flows in an ad-hoc network without QoS support.
OTHER QUALITY OF SERVICE MODELS ON MANETS

The SWAN [7] model has been developed by the Comet team at Columbia University. The model differentiates traffic into real-time UDP traffic and best effort TCP traffic. It is a stateless and fully distributed model that provides soft QoS assurances to real-time traffic. It uses admission control for real-time traffic, rate control of TCP traffic and ECN congestion control mechanisms to ensure that real-time packets meet QoS bounds. Each node comprises an admission controller that maintains information about the status of the outgoing link in terms of the available bandwidth and amount of congestion. It does this by promiscuously listening to all packet transmissions within its range. The admission controller located at the source node sends a probe message toward the destination when a new real-time flow requires servicing. The probe message returns carrying the value of the bottleneck bandwidth along the path. If this value is greater than the requirements plus a threshold value, the flow is admitted; otherwise it is rejected and marked as best-effort. All TCP flows are considered as best-effort. The best-effort traffic passes through a rate-controller that shapes the traffic according to the rate based on the feedback from the MAC layer. The admitted real-time traffic bypasses the rate controller and has a scheduling priority over best-effort traffic. The admitted real-time flows only have soft QoS assurances, so that some of the flows may be dropped or downgraded to best effort if network traffic conditions change due to rerouting of traffic.

INSIGNIA [5,9] is an in-band signaling system that supports adaptive reservation-based services in ad hoc networks. Thus all the control information is carried within the header of the data packet itself, without the need of a separate control channel. The INSIGNIA framework has also been developed by the Comet group at Columbia University.

The signaling system supports a number of protocol commands that drive fast reservation, fast restoration and end-to-end adaptation mechanism. These commands are carried in-band with the data and encoded using the IP option field in datagrams. The in-band information is snooped as data packets traverse intermediate nodes/routers and used to maintain soft-state reservations in support of flows/micro-flows. To establish reservation-based flows between source-destination pairs, source nodes initiate fast reservations by setting the appropriate fields in the INSIGNIA IP option field before forwarding packets. Reservation packets (i.e. data packet with the appropriate IP option set) traverse intermediate nodes, executing admission control modules, allocating resources and establishing soft-state reservation at all intermediate nodes between source-destination pairs. The reservations need to be periodically refreshed by the packets of the flows. In the event of a change in the path resulting from movement of the nodes, the first packet along the new path makes fresh reservations along this path thereby performing a fast restoration. Reservations made along the old path are removed on a timeout. Flows in the network are expected to be adaptive to bandwidth availability.

A flow that was allocated a MAX amount of bandwidth initially could be downgraded to a MIN amount or even to best-effort in the event of rerouting of a flow or if network conditions change. The source node continues to send packets with the reservation bit set until the destination node completes the reservation setup phase by informing the source node of the reservation status using a QoS reporting mechanism.

ASAP [4] is an adaptive reservation QoS protocol. It has a two-phase reservation scheme and a simple signalling system. When a flow requires service, it is soft-reserved first and later hard reserved. Soft reservation is more like a promise that the resources will be allocated to the flow, however they can still be used by another flow until the time they are hard-reserved to the flow. The two-phase reservation mechanism helps in not wasting resources by dedicating them to a flow during route discovery whilst other flows need the resource. When the route has been established the resources are then hard-reserved to the flow. Hard-reservation is an actual dedication of resources to a particular flow. ASAP got an adaptation method that will allow some flows to release bandwidth if the allocated bandwidth is more that their required minimum. It also has a route repair mechanism that allows self-healing by finding alternative routes when a route is broken, which happens frequently in wireless networks. ASAP adaptation helps to stabilise the throughput of flows in wireless ad hoc networks as shown by the simulations we ran on ASAP, Figure 2. This gives a better quality as compared to the case shown by Figure 1.

Figure 2 Graph showing throughput for two flows that are subjected to bandwidth adaptation in ASAP.
This shows that bandwidth management is very important in mobile ad hoc networks. However when the number of nodes increases and they are very mobile then the number of dropped packets also increases. There is a need to monitor the admission of more flows into the wireless network in relation to the available bandwidth in order to curb this problem. It is important to admit less traffic, which can then be transmitted with enough quality assurance, rather than admit a large number of flows which are barely useful in terms of multimedia quality. It is in this interest that researchers are designing a new Quality of Service model to fill this gap.

THE PROPOSED FRAMEWORK

The main functionalities of resource management in Mobile Ad Hoc Networks (MANETs) are Resource Estimation, Admission Control and Resource Reservation. These functionalities should be handled in a way that avoids the wastage of resources and interference with other on-going communications. In this paper we propose a traffic management framework, which is composed of an admission control module and bandwidth adaptation, to achieve Quality of Service in MANETs. The traffic is differentiated into distinct classes depending on their quality of service demands. The framework gives the highest priority class preemptive preference over lower priority classes. Real-time traffic’s quality degrades appreciably below a specified throughput and so bandwidth sharing is inappropriate. When the available bandwidth is not enough, then the network should not admit new flows, otherwise it will interfere with on-going flows whilst it does not meet its own minimum requirements.

However, the adaptation module must degrade the least priority flows to their minimum required bandwidth in order to free resources for the new flow. The module must be able to calculate the amount of bandwidth to be released by low priority flows before degrading them. If the highest priority flow asks for resources when the remaining resources are not enough, then we have to pause some of the least priority traffic to release resources.

In case of network congestion notification, we can adapt the ongoing flows to reduce the allocations (throttle) until the channel is no longer congested. If congestion persists then the least priority flow should be paused. After a random back-off time a source with a throttled or paused flow can attempt to increase or re-admit the traffic flow. For this study, we assume all flows require a minimum level of service below which the flow cannot be throttled. If this threshold has been reached, we have to pause flows to avoid congestion. To avoid reducing reservations for multiple flows in response to mobility-induced congestion, we pause the flows depending on their priorities.

The least priority is paused only if all the flows have reduced their reservations with the network remaining congested. After this our system goes into, what we call, panic mode. In the panic mode the network should start stopping the flows starting with the least priority flow. More flows will be stopped until the system state has been restored to a non-congested state.

![Diagram showing all six important components](image)

**Figure 3:** The proposed theoretical QoS Framework Block Diagram showing all the six important components.

PERFORMANCE MODELING

The framework was simulated in C++ code. The code consists of several functions which are called to perform different tasks including computing the available bandwidth. A flow can be identified with the following attributes:

1. Flow ID - to uniquely identify all the flows in the network and also to keep track of the flows which were reduced to required minimum during bandwidth adaptation.
2. Flow priority - to resemble different real-time traffic, with the flow with the lowest value having the highest priority. These values are important in the Bandwidth Adaptation module when determining which flow(s) to reduce allocation in order to admit a new flow. Flows with the least priorities are reduced first.
3. Required Minimum Bandwidth (RQmin) – This is the minimum required bandwidth requirements of a flow in the network
4. Releasable Bandwidth (RLBW) – This is the difference between the reserved bandwidth and the required minimum. The bandwidth that can be released for other new flows to be admitted.

All these values were randomly generated for experimental studies only. However in future we intend to use ns-2 traffic sources when we manage to embed our code into the ns-2 simulator. An array of up to 1000 flows is defined and has been used for different flows in the simulations. The Channel Capacity is assumed to be 2Mb. The available bandwidth at any given time will be the difference between the Channel Capacity and the Consumed Bandwidth (i.e. the bandwidth consumed by all the flows in the network).
RESULTS

This part shows some performance modeling of the bandwidth management system on mobile ad hoc networks. We show how the network admits traffic up to a time when all new flows are rejected. We also show how by adapting reservations downwards bandwidth can be released to allow admission of new flows. We also test the model under congestion state to gauge its effectiveness in selective bandwidth adaptation and flow pausing.

There are three things that can happen when a call for admission is made for a new flow:

No Adaptation and Admission of flow – This happens when the bandwidth requirements of the new flow are met by the available bandwidth and there is no need for Adaptation, thus the flow is automatically admitted into the network. Figure 4 shows a sample of the output of a particular simulation, which had 42 flows initially and the consumed bandwidth, CBW, was 1.476191 and thus the available bandwidth was 0.523809 (the difference between the Channel Capacity - 2Mb -, and the consumed bandwidth). The releasable bandwidth, RLBW, from other flows already in the network is 0.460565. This value corresponds to the sum of all releasable bandwidth by each flow, which is the difference between a flow’s required bandwidth and its required minimum. The output also shows the new packet data, which includes the required bandwidth of 0.152905, priority of 1 and a minimum requirement of 0.058824. The flow is admitted automatically without any need for adaptation since the available bandwidth is sufficient to cater for the bandwidth requirements of the new flow.

Adaptation and Admission of flow – When the bandwidth requirements of the new flow exceed the available bandwidth, there is a need for reducing allocations of other flows already in the network and then admitting the flow when the requirements are met. Figure 5 shows some highlights of the adaptation process in which other flows have their allocations reduced to their minimum requirements based upon the priorities and then the new flow will be admitted once sufficient bandwidth is available.

Adaptation and no Admission of flow - When the bandwidth requirements of the new flow exceed the available bandwidth and adaptation is done but the bandwidth requirements cannot be met, even after all flows have been reduced to their working minimum, then the flow is denied. A sample output of a denied flow is shown in Figure 6.

Table 1 shows the results from one simulation showing all the flows in the network and the amount of bandwidth each network is capable of releasing. In Table 2, in the first simulation, a total of 19 flows out of the 22, which were initially in the network, were reduced and eventually the flow was admitted after the bandwidth requirements were met. The captured statistics were used in the analysis of the adaptation process so as to verify whether the adaptation process is useful within a network. This will help in coming up with a proposal or recommendation of the system based upon the analysis of these results. The results that can be derived from the simulations include calculating the success rate of including bandwidth adaptation in the resource reservation scheme and calculating the success rate of the Bandwidth Adaptation process itself, i.e. what is the success factor that a simulation that invokes the Bandwidth Adaptation module leads to the admission of a new flow? We also study the relationship and the effect of having less or many flows in the network on the admission of a new flow and on the number of flows that can be reduced to their working minimum.

Using the statistics, we estimate the success rate of a scheme with Bandwidth Adaptation $R$ as:

$$ R = \left( \frac{S_{\text{admitted}}}{S_{\text{total}}} \right) \times 100 \quad (1) $$

where $S_{\text{admitted}}$ is the value of simulations with flows Admitted after Adaptation and $S_{\text{total}}$ is the total number of simulations. We found the success rate $R$ to be 73.3% for a scheme with Bandwidth Adaptation. However for a scheme, without Bandwidth Adaptation and using the same formula, the success rate is only 40%.

We estimate the success rate of the Bandwidth Adaptation process as the fraction number of simulations with flows admitted after adaptation of the total number of simulations with adaptation.

$$ R_{\text{adapt}} = \left( \frac{S_{\text{admitted}}}{S_{\text{adapted}}} \right) \times 100 \quad (2) $$

where $R_{\text{adapt}}$ is the success rate of the adaptation process and $S_{\text{adapted}}$ is the total number of all flows that went through adaptation. We found the rate to be 60%.

Table 2 highlights simulations with flows that were admitted after adaptation and the number of flows already in the network, which were reduced to their required minimum. This helps in noting the relationship and the effect of having less or many flows in the network on the admission of a new flow. We plot a graph, Figure 7, of the number of flows in simulations against the number of flows reduced in order to visualize the effect of network size (represented by number of flows in network) on the admission of a new flow.

DISCUSSION

The results show that the bandwidth requirements of a flow have an effect on whether the flow will be admitted or not. Any flow with a bandwidth requirement above 0.75Mb cannot be admitted unless it has passed through the Adaptation module and the available bandwidth is increased to levels above the required bandwidth. It therefore means that the introduction of a flow, with requirements more than 0.75Mb, into a network results in the flow being denied admission unless other mechanisms like Adaptation are implemented to increase the available bandwidth in the network. Thus, this scheme is efficient in enhancing the chances of admission of flows with greater requirements.
The proposed scheme relies on bandwidth adaptation for maximum efficiency of the system and the adaptation is highly dependent on the Available Bandwidth and the Requested Bandwidth by the new flow. Using the estimates for the success rate of a scheme with Bandwidth Adaptation calculated using the formula in equation (1) we can clearly see the role played by the adaptation process in increasing the available bandwidth in the network so that the bandwidth request of the new flow can be met and thus, ensuring that the success rate of admission of a flow with bandwidth requirements above 0.75Mb is enhanced by 73%. Therefore, comparing with the statistics calculated for a scheme without bandwidth adaptation, 40%, it is evident that bandwidth adaptation enhances the system efficiency in the admission of more flows.

The size of a network, represented by the initial number of flows in the network, greatly impacts the number of flows, whose requirements will be reduced to required minimum. The results show that the bigger the network size, the bigger the set of flows that will be reduced to required minimum. However, this might have adverse negative impacts on the performance and output of the flows reduced especially real-time traffic like audio and video streaming. In real life simulations to be carried out using the NS-2 Simulator, the size of the network need to be put into consideration before introducing a new traffic flow into the network in order to control congestion, dropping of packets and other problems that might be caused by the reduction of bandwidth requirements of flows to required minimum.

**RECOMMENDATIONS AND CONCLUSIONS**

It is difficult, at this point, to make a qualitative comparison of this scheme with other schemes like ASAP, SWAN and INSIGNIA because this scheme is not adequate since it needs to be embedded in an ad hoc routing protocol, then re-implemented and simulated in the NS-2 Simulator. As a result, this scheme will only be compared against a scheme without Adaptation in order to assess the efficiency of the Bandwidth Adaptation process, since it is the novel part of the research.

Though the proposed scheme might pose implementation problems in real life situations, from the experimental results obtained and calculated, we believe the benefits of the scheme outweigh the weaknesses. Thus, we recommend adopting the scheme, embedding it in an ad hoc routing protocol and integrating it with other reservation components as outlined in the Methodology.

The scheme has the potential of coming up with efficient and realistic reservations, which are comparable to those of other frameworks, like ASAP and SWAN.

<table>
<thead>
<tr>
<th>Number of flows</th>
<th>42</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Bandwidth</td>
<td>0.523809</td>
</tr>
<tr>
<td>CBW</td>
<td>1.476191</td>
</tr>
<tr>
<td>RLBW</td>
<td>0.460565</td>
</tr>
</tbody>
</table>

**Figure 4:** Enough resources available, no adaptation and packet is admitted

<table>
<thead>
<tr>
<th>Number of flows</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Bandwidth</td>
<td>0.333333</td>
</tr>
<tr>
<td>CBW</td>
<td>1.666667</td>
</tr>
<tr>
<td>RLBW</td>
<td>1.197917</td>
</tr>
</tbody>
</table>

**Figure 5:** Resources are not enough so there is adaptation and packets are admitted

<table>
<thead>
<tr>
<th>Number of flows</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Bandwidth</td>
<td>0.791667</td>
</tr>
<tr>
<td>CBW</td>
<td>1.208333</td>
</tr>
<tr>
<td>RLBW</td>
<td>0.407552</td>
</tr>
</tbody>
</table>

**Figure 6:** Resources are not enough so the system tries to adapt but still there is no admission
Table 1 Results showing adaptation process in a single simulation run.

<table>
<thead>
<tr>
<th>FLOW ID</th>
<th>RSW BW</th>
<th>REQ MIN</th>
<th>PRIORITY</th>
<th>RL BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.3906</td>
<td>0.3906</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.01953</td>
<td>0.01953</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.01953</td>
<td>0.01953</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0.07813</td>
<td>0.07813</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0.3906</td>
<td>0.3906</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0.3906</td>
<td>0.3906</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.3906</td>
<td>0.3906</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0.3906</td>
<td>0.3906</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0.01953</td>
<td>0.01953</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.3906</td>
<td>0.3906</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0.3906</td>
<td>0.3906</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0.01953</td>
<td>0.01953</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0.3906</td>
<td>0.3906</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0.01953</td>
<td>0.01953</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0.01953</td>
<td>0.01953</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0.01953</td>
<td>0.01953</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0.09091</td>
<td>0.05859</td>
<td>0</td>
<td>0.03232</td>
</tr>
<tr>
<td>16</td>
<td>0.09091</td>
<td>0.7813</td>
<td>0</td>
<td>0.01278</td>
</tr>
<tr>
<td>17</td>
<td>0.09091</td>
<td>0.05859</td>
<td>0</td>
<td>0.03232</td>
</tr>
<tr>
<td>0</td>
<td>1.01961</td>
<td>1.01961</td>
<td>0</td>
<td>0.1022</td>
</tr>
</tbody>
</table>

Table 2 Results showing simulations with flows admitted with Adaptation and the number of flows reduced to working minimum in each case.

<table>
<thead>
<tr>
<th>NUMBER OF FLOWS</th>
<th>AVAIL BW</th>
<th>REQ BW</th>
<th>FLOWS REDUCED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36</td>
<td>0.6944</td>
<td>0.9174</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1.4667</td>
<td>0.7645</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>0.7</td>
<td>0.9174</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>0.5455</td>
<td>1.0703</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>0.3333</td>
<td>1.2232</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>1</td>
<td>1.4526</td>
</tr>
<tr>
<td>22</td>
<td>37</td>
<td>0.6486</td>
<td>0.9174</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>0.7</td>
<td>0.7645</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>0.5417</td>
<td>0.8409</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>0</td>
<td>0.3823</td>
</tr>
<tr>
<td>26</td>
<td>15</td>
<td>0.6607</td>
<td>0.6881</td>
</tr>
<tr>
<td>29</td>
<td>19</td>
<td>0.6316</td>
<td>0.7645</td>
</tr>
</tbody>
</table>
REFERENCES


8. Fu Z, Meng X, and Lu S. TCP friendly rate adaptation for multimedia streaming in mobile ad hoc networks,


BIOGRAPHY

BENNY M. NYAMBO obtained the Bachelor of Science Honours degree in Physics in 1999 and MSc in Applied Physics in 2002, both from the University of Zimbabwe, Harare, Zimbabwe. He is currently studying towards a PhD in Computer Science at the Hasselt University, Belgium. He has been a lecturer in Computer Science and the University of Zimbabwe since 2003. He has research interests in performance modelling of communication networks and in embedded systems.

GERRIT K. JANSENS 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 PhD. 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. His main research interests include the development and application of operations research models in production and distribution logistics.

WIM LAMOTTE obtained his PhD in computer science in 1998 at Hasselt University, Belgium, where he is now a full-time professor teaching courses in networking and multimedia. His current research interests include networked virtual environments, computer networks and networked multimedia. He leads the Multimedia and Communication Technology research group of the Expertise Centre for Digital Media, specializing in Networked Virtual Environments. He is member of IEEE and ACM.
MODELING AND SIMULATION OF ADVANCED FLOATING BODY Z-RAM MEMORY CELLS

Viktor Sverdlov and Siegfried Selberherr

Institute for Microelectronics
Technische Universität Wien
Gussmannstrasse 27–29
1040 Vienna, Austria
Email: {sverdlov|selberherr}@iue.tuwien.ac.at

KEYWORDS
Floating body memory, TCAD simulations, scalability

ABSTRACT

A modeling approach to study advanced floating body Z-RAM memory cells is developed. In particular, the scalability of the cells is investigated. First, a Z-RAM cell based on a 50 nm gate length double-gate structure corresponding to state of the art technology is studied. A bi-stable behavior essential for Z-RAM operation is observed even in fully depleted structures. It is demonstrated that by adjusting the supply source-drain and gate voltages the programming window can be adjusted. The programming window is appropriately large in voltage as well as in current.

We further extend our study to a Z-RAM cell based on an ultra-scaled double-gate MOSFET with 12.5 nm gate length. We demonstrate that the cell preserves its functionality by providing a wide voltage operating window with large current differences. An appropriate operating window is still observed at approximately 25-30% reduced supply voltage, which is an additional benefit of scaling. The relation of the obtained supply voltage to the one anticipated in an ultimate MOSFET with quasi-ballistic transport is discussed.

INTRODUCTION

Standard DRAM cell scaling is hampered by the presence of a capacitor which is difficult to reduce in size. Recently, a revolutionary concept of a DRAM memory cell based on a transistor alone was introduced [1–3]. The ultimate advantage of this new concept is that it does not require a capacitor, and, in contrast to traditional 1T/1C DRAM cells, it thus represents a 1T/0C cell named Z (for zero)-RAM. While keeping all advantages of the first Z-RAM generation, the most recent generation of Z-RAM cells [4] is characterized by a significantly enlarged programming window and much longer retention times.

With CMOS downscaling continuing the question obviously arises whether a Z-RAM cell is also scalable. The goal of our study is to demonstrate that a Z-RAM cell preserves its functionality and remains operational for scaled MOSFETs.

In order to reach this goal, several issues must be addressed. Multi-gate FETs and finFETs are the most promising candidates for the upcoming CMOS MOSFETs beyond the 22nm technology node. The functionality of both generations of Z-RAM cells on partially depleted SOI structures was recently demonstrated [4]. With channel length reduced, maintaining control over the channel becomes increasingly challenging, and several measures must be taken to preserve it. Apart from improving electrostatic control by downsampling oxide thickness, the channel width can be reduced. This is achieved by artificially confining carriers within an ultra-thin silicon film. Due to the small dimensions of the silicon body there will be only few impurities inside. This results in unacceptably large threshold voltage fluctuations [5]. Fully depleted double-gate MOSFETs with undoped intrinsic silicon body are perfectly functional [6, 7]. They preserve a good channel control, reasonable DIBL, large $I_{on}/I_{off}$ ratio, and gain down to a channel length as short as 5 nm [8]. It is not clear, however, whether a Z-RAM cell based on a fully depleted scaled MOSFET would be operational.

Z-RAM cell functionality is based on charging the channel body with the majority carriers generated due to impact ionization. Therefore, for Z-RAM operation, namely for writing, the source-drain voltages are higher than for CMOS logic. An important question is whether this voltage can be reduced while scaling the device down. We demonstrate that, as for CMOS devices, this is generally true. Calculated voltages for scaled Z-RAM cells are around 1.4 V. This value is also in agreement with the writing voltage in a quasi-ballistic MOSFET with an ultra-short channel, which is considered as a good candidate for an ultimate MOSFET [5]. The value is higher than the projected $V_{DD}$’s for upcoming MOSFETs. However, the current prototypes of a Z-RAM cell operate now at 2.2 V [4]. Therefore, a decrease in supply voltage to 1.4 V is significant.

Simulated structures and models are described in the next section. Then results are presented and analyzed.
STRUCTURES

For our analyses we have designed two double-gate structures. One structure has a gate length of 50 nm and a lightly doped \( N_A = 10^{15} \text{ cm}^{-3} \) Si body of 10 nm thickness. We have used metal gates with mid-gap work function and oxide with equivalent thickness of 2 nm. Source and drain extensions are heavily doped to \( N_D = 10^{20} \text{ cm}^{-3} \) in order to provide enough injected electrons. This structure corresponds to the current technology node [9].

The scaled double-gate structure has a gate length of 12.5 nm and a lightly doped Si body of 3 nm thickness. An oxide with an equivalent thickness of 1 nm is chosen. The source and drain extensions are heavily doped to \( N_D = 10^{20} \text{ cm}^{-3} \).

The analyses were performed with the MINIMOS-NT device simulator [10]. Impact ionization is essential to the functionality of a Z-RAM cell. Electron-hole recombination is very important as antagonistic mechanism. Similar parameters for impact ionization and for recombination are used for both structures. Band-to-band tunneling was also included with a standard model [10].

RESULTS

The results for current calculations as function of the gate voltage for 50 nm double-gate structure are shown in Fig. 1 and Fig. 2. At high positive gate voltages the current values do not depend on the gate voltage scan direction shown by left and right triangles.

For negative gate voltages the situation is completely different. In a forward scan direction for the gate voltage, the current stays low for both values of the source-drain voltage until a certain critical value is reached. This part of the \( I - V_G \) corresponds to the subthreshold regime. Due to negligible DIBL in a 50 nm double-gate structure, the current dependence in the subthreshold regime is similar for both values of the source-drain current.

As soon as a critical current value is reached, the source-drain current rapidly increases by several orders of magnitude. The current keeps increasing for positive gate voltage values.

In a reverse gate voltage scan, the current first slowly decreases. For positive gate voltages, the current takes exactly the same values as for the forward scan. Therefore, the \( I - V_G \) curve is completely reversible for both values of the source drain voltage, as already mentioned. However, the current value does not decrease sharply for moderately negative values of gate voltages, although it keeps slightly decreasing. The current values at the reverse scan remain several orders of magnitude higher than the values for the forward scan. The relatively large current value is maintained down to \( V_G = -2 \text{ V} \) for \( V_{SD} = 2.0 \text{ V} \), where it abruptly decreases by several orders of magnitude. Thereby the MOSFET is turned back into the subthreshold regime, completing the hysteresis loop. Indeed, the previous current value cannot be reached just by inverting the scan direction. Instead, if one now increases the gate voltage, the current will follow the lower subthreshold branch until the critical
current value is reached at \( V_G = -0.6 \) V. The point with relatively high current at \( V_G = -2 \) V can only be reached by inverting the gate voltage scan after the high current value was achieved at positive gate voltages. Interestingly, if we increase the source–drain voltage to \( V_{SD} = 2.2 \) V, the abrupt transition to the subthreshold regime during the reverse voltage scan cannot be observed for technically relevant negative gate voltage values. A similar behavior is observed when a parasitic bipolar transistor turns on in floating body SOI structures [11], which usually is considered as undesirable.

The two different current states corresponding to the same drain and gate voltages, seen also on \( I-V_{SD} \) characteristics shown in Fig. 2 are essential for Z-RAM functionality [4]. We are now analyzing the physical reasons for these two different current states.

Fig. 3 displays the potential profile from the source to the drain electrode cut in the middle of the silicon body, for two different current states corresponding to the same source and drain voltage. In the low current state the potential has a large barrier under the gate preventing the current flowing from the source to the drain. On the contrary, in the high current state the potential is nearly flat in the source-gate region, and the transistor is opened. Such a difference in potential profiles is due to different charge distributions in the system shown in Fig. 4. In the state with low current the electron concentration in the channel is small and he majority carrier concentration (holes) is small too (Fig. 4). In the state with high current one naturally has a higher electron concentration in the channel.

More important, the hole concentration in the channel has also increased as shown in Fig. 4. The hole concentration in the channel is higher close to the gates (Fig. 5), in agreement with [2]. Holes are generated close to the drain region due to impact ionization. The electric field drives generated holes in the channel region where they accumulate. This accumulated positive charge pins down the conduction band to the potential in the source and opens the transistor. Holes recombine with electrons primarily via the Shockley-Read-Hall mechanism. Excess holes visible in Fig. 4 flow into the source region. Their extra positive charge is compensated by additional electrons, resulting in a slightly higher electron concentration than the equilibrium concentration determined by \( N_D = 10^{20} \) cm\(^{-3}\). The nature of the two current states analyzed above allows to determine conditions, when the transitions between them occur. One important ingredient is impact ionization which is usually characterized by the multiplication factor \( M > 1 \). The positive feedback loop is activated when the collector current is larger than the base current. Because the hole base current is proportional to \( M - 1 \), the positive feedback corresponds to the condition [11]

\[
\beta_F (M - 1) > 1,
\]

where \( \beta_F \) is the common-emitter current gain. The increase of the drain current is triggered by the positive feedback, saturating when the transistor opens. If we now reduce the gate voltage, an increasing number of holes must be stored under the gate to compensate.
figure 5: contour plot in logarithmic scale of the hole distribution in the MOSFET channel for the high current state. Close to the gates the hole concentration is higher, in agreement with [4].

the gate voltage decrease and keep the transistor open. This results in an increased recombination rate that reduces $\beta_F$. At large negative gate voltages the condition (1) cannot be fulfilled. The positive feedback loop breaks, which results in a sudden decrease of the current. The number of generated holes drops. Their concentration under the gate reduces and they cannot screen the gate potential. This leads to a potential barrier increase which further reduces the current. The process stops when the transistor is closed.

The consideration above can explain the $I-V_G$ behavior at $V_{SD} = 2.0$ V for the reverse gate voltage scan. For the forward gate voltage scan the transistor stays in closed state for higher gate voltages than for the reverse scan. The reason is the absence of current in the closed state. However, due to an exponential current increase in the subthreshold regime at the gate voltage close to zero, the current reaches a critical value after which the positive feedback loop leading to transistor opening activates. The critical current values only slightly depend on drain voltage, due to a dependence of $\beta_F$ on drain voltage.

We have demonstrated that a programming window, which is formed by the two current values and the two gate voltage values when switching appears, is sufficiently large for stable Z-RAM operation on 50 nm double-gate transistors. We now present simulations of a double-gate structure with 12.5 nm gate length. Results of $I-V_G$ calculations shown in Fig. 6 and Fig. 7 clearly display a hysteresis behavior similar to that observed for a 50 nm MOSFET. For all considered source-drain voltages the transition to the high current state appears at slightly negative gate voltages. For the reverse scan the transition to the low current state is observed at large negative gate voltages for $V_{SD} = 1.6$ V, while for $V_{SD} = 1.2$ V the hysteresis has nearly disappeared. For $V_{SD} = 1.4$ V, the transition to the low current state occurs at $V_G = -0.6$ V. This results in a relatively large programming window sufficient for successful Z-RAM cell operation. It is thus demonstrated that a Z-RAM cell built on a scaled double-gate MOSFET with 12.5 nm gate length preserves its functionality.

DISCUSSION

The $I-V_G$ behavior for a 12.5 nm gate length MOSFET looks analogous to the behavior of a 50 nm MOSFET. One difference between the results is that the current density for a thinner and shorter double-gate structure is nearly an order of magnitude smaller. However, this is not a substantial limitation, because the important criterion for Z-RAM functionality is the difference between the two values of current in the two different current states, which is still several orders of magnitude for a 12.5 nm double-gate structure.

Another important difference is that the supply voltages are 25-30% smaller for a Z-RAM based on a scaled MOSFET. The obtained substantial decrease in supply voltage is comparable with the anticipated decrease of $V_{DD}$ for scaled logic devices. We should add, however, that the impact ionization model used in the simulations depends on the local field only. When the channel length is reduced, the local field in the channel at the drain end is expected to increase. Therefore, the local field impact ionization model can overestimate impact ionization. Another potential limitation of the applica-
Figure 7: $I - V_{SD}$ in a logarithmic scale for a 12.5 nm double gate MOSFET with silicon body thickness 3 nm, for different gate voltages $V_G$. Hysteresis behavior is clearly observed.

bility of our approach is that in scaled devices transport becomes quasi-ballistic, and the impact ionization models used in Monte Carlo simulations of hot carrier transport [12] should become valid. These models are characterized by threshold energies above which impact ionization starts, with the lowest threshold of 1.2 eV. It was recently argued that due to the presence of energetic carriers in an injected distribution substantial impact ionization can be present even, when the source-drain voltage is smaller than the threshold [13]. However, because of the gap increase due to size quantization in a 3 nm silicon film, we believe that 1.2 eV is a fair estimate of a Z-RAM cell supply voltage, which is consistent with $V_{SD} = 1.4$ V obtained in our simulations.

CONCLUSION

We have shown that a Z-RAM cell built on a scaled double-gate MOSFET preserves its functionality by providing a wide voltage operating window with large current differences. We also predict a decrease in the supply drain-gate voltage to 1.2-1.4 V, which is about 25-30% smaller than in current prototypes.

REFERENCES


DISTRIBUTION OF THE SUCCESSFUL AND BLOCKED EVENTS IN THE M/G/1 RETRIAL QUEUE

J. Amador
School of Statistics, Complutense University of Madrid, 28040 Madrid, Spain
E-mail: jamador@estad.ucm.es

KEYWORDS
Queueing, Retrals, Blocked events, Successful events.

ABSTRACT

It's dealt with a single retrial queue with exponential repeated attempts and general distribution for the service times. The main difference between a retrial queue and a standard waiting line is that the retrial group performs as an invisible queue, and then the original flow of primary arrivals and the flow of repeated attempts become undistinguished. Having this into account, it has been analyzed the distribution of the following four descriptors, referred to a busy period: number of successful retrals, blocked retrals, successful primary arrivals and blocked primary arrivals.

1. INTRODUCTION

This work is focussed on a branch of the queuing theory, retral queues, which is characterized by the following feature: a customer who cannot receive service leaves the service area but after some random time returns to the system again. As a consequence, repeated attempts for service from the retral pool of unsatisfied customers are superimposed on the ordinary stream of arrivals of first attempts.

The main goal is to investigate the distribution of the following performance descriptors of the customer's behaviour: the successful retrals, the successful arrivals and the blocked retrals. It will be showed that the number of blocked primary arrivals amounts to the number of successful retrals referred to a busy period. Hence, the three measures provide a full description of what is relevant in practice in order to distinguish between primary arrivals and repeated attempts behaviours. Attention is paid to methods for the computational analysis of the performance measures under study. To this end, direct methods are proposed for the computation of the probability mass functions instead of using an alternative approach based on generating functions. The main advantage of the direct approach is to avoid the numerical inversion of the generating functions.

These descriptors are studied for the M/G/1 retrial queue. It's a semi-Markovian model observed at service completion epochs (Falin and Templeton 1997).

In the next section the queuing model under consideration is described and the four performance descriptors are defined. In subsequent Sections 3-5, it's dealt with the computational analysis of the three proposed descriptors. Finally, in Section 6 some numerical results and conclusions are presented.

2. THE QUEUING MODEL AND THE PERFORMANCE DESCRIPTORS

M/G/1 retrial queue is next described. Customers arrive according to a Poisson process of rate λ. If a primary customer finds the server free, he immediately occupies it and he leaves the system after service. The service times are general with probability distribution function B(x) (B(0) = 0), Laplace-Stieljes transform β(s) and moments βk, for k ≥ 1. On the other hand, any arriving customer who finds the server busy leaves the service area and joins a retrial group called orbit. The policy of access from the orbit to the server is governed by an exponential law with rate jµ, given that the number of customers in orbit is j ≥ 0. It's assumed that the flow of primary arrivals, intervals between repeated attempts and service times are mutually independent.

At any time t the state of the system can be described by X(t) = (C(t), N(t)), where C(t) is the state of the server (0 if it's free and 1 if it's busy) and N(t) is the number of customers in orbit at time t. It's assumed that ρ = λβ1 is lesser than 1, so the system is stable and the limiting probabilities

\[ p_i = \lim_{t \to \infty} P[C(t) = i, N(t) = j] \]

exist and are positive (Falin and Templeton 1997; Artalejo and Gómez-Corrals 2008).

Now, the performance descriptors are defined under a busy period: \( R^* \) is the number of successful retrals, \( A^* \) is the number of successful arrivals, \( R^h \) is the number of blocked retrals and \( A^h \) is the number of blocked arrivals. Since during a busy period the number of blocked retrals and the number of blocked arrivals are coincident (Amador and Artalejo 2007), it's only needed to study one of them, \( R^* \) for example. The exact expressions for the mean values of these descriptors are found (Amador and Artalejo 2008; Artalejo and Lopez-Herrero 2007; Artalejo and Gómez-Corrals 2008).
3. THE NUMBER OF SUCCESSFUL RETRIALS

Recursive equations for the exact computation of the probability mass function \( P[R^r = r] \) are derived, for \( r \geq 0 \).

Let \( x^r_i (r) \) be the probability of having exactly \( r \geq 0 \) successful retrials during the remaining busy period, given that a service time has just been completed leaving \( i \) customers in orbit, for \( 0 \leq i \leq r \). Then, it is have

\[
P[R^r = r] = \sum_{j=0}^{r} c_j x^r_j (r), \quad r \geq 0,
\]

where \( c_k \), for \( k \geq 0 \), is the probability that \( k \) arrivals occur during a service time.

The probabilities \( x^r_i (r) \) satisfy an upper triangular system (Amador and Artalejo 2008). It is suitable for recursively getting the probabilities \( x^r_i (r) \) and, consequently the probability mass function of \( R^r \) is computed.

4. THE NUMBER OF SUCCESSFUL ARRIVALS

To study the number of successful primary arrivals the \( M/G/1 \) retrial queue with infinite orbit is approximated by the truncated model with orbit capacity \( K \geq 1 \) (Falin and Templeton 1997; Wilkinson 1956). Blocked customers finding the state \((1,K)\) upon arrival are lost. It’s necessary to deal with a truncated system; otherwise, it would have to be solved an infinite system which has no known solution.

Let \( x^a_i (a) \) be the probability of having \( a \geq 0 \) successful arrivals during the remaining busy period, given that a service time has just been completed leaving \( i \) customers in orbit, for \( 0 \leq i \leq K \). Then,

\[
P[A^r = a] = \sum_{j=0}^{K-1} c_j x^a_j (a) + \left( 1 - \sum_{j=0}^{K-1} c_j \right) x^K_k (a), \quad a \geq 0.
\]

The probabilities \( x^a_i (a) \) are the solution of a linear finite system (Amador and Artalejo 2008). So, once this system is solved, the probability mass function of \( A^r \) for the truncated model is calculated.

5. THE NUMBER OF BLOCKED RETRIALS

The number of blocked retrials taking place during a given service time depends on the arbitrary number of blocked primary arrivals occurring during the service time in progress. Thus, it seems difficult, or even impossible, to obtain the exact distribution of \( R^b \). Two methods for the computation of the probability mass function of \( R^b \) are proposed for the model with finite capacity \( K \geq 1 \). The aim of the first approximation is to present a tractable model for accurately representing the service times. To this end, it’s assumed that service times follow a \( PH(\alpha, T) \) distribution of order \( s \), where the row vector \( \alpha \) gives the initial phase distribution and matrix \( T \) governs the infinitesimal phase rates. Phase type distributions form a versatile family of probability distributions.

The state of the \( M/PH/1/K \) queue with retrials is described by the Markov chain \( Y(t) = (K(t), N(t)) \), where \( K(t) \) is the phase of the service in progress. Note that \( K(t) = 0 \) indicates that the server is idle.

Conditioning on the phase of the first arrival of the busy period, it’s got

\[
P[R^b = r] = \sum_{k=1}^{s} \alpha_k y^b_k (r), \quad r \geq 0,
\]

where \( y^b_k (r) \) is the probability of having \( r \) blocked retrials during the remaining busy period, given that the current system state is \((k,j)\), for \( 0 \leq k \leq s \) and \( 0 \leq j \leq K \). These unknowns \( y^b_k (r) \) satisfy a finite block tridiagonal system (Amador and Artalejo 2008).

If the wish is to deal with any arbitrary service time, not necessarily of \( PH \) type, it can be done with the help of the following approach.

With the help of Kronecker’s function

\[
\delta_{ab} = \begin{cases} 
1 & \text{if } a = b, \\
0 & \text{otherwise}, 
\end{cases}
\]

the probabilities of \( R^b \) can be expressed

\[
P[R^b = r] = \delta_{r,0} c^0_0 + (1 - \delta_{r,0}) \sum_{k=0}^{K-1} \sum_{l=0}^{r} c^l_k x^b_k (r-k)
\]

\[
+ \sum_{k=0}^{r} d^k_{i-k} x^b_k (r-k), \quad r \geq 0,
\]

where \( c^l_k \) is the probability that \( j \) primary arrivals and \( k \) retrials occur during a service time, given that immediately after the beginning of the service time the orbit size is \( i \), for \( 0 \leq i \leq K \), \( j \geq 0 \) and \( k \geq 0 \). \( d^k_{i-k} = \sum_{m=0}^{\infty} c^m_i \) for \( 0 \leq i \leq K \) and \( k \geq 0 \). The quantities \( x^b_i (r) \) denote the probability of having \( r \) blocked retrials during the remaining busy period, given that a service time has been completed leaving \( i \) customers in orbit, for \( 0 \leq i \leq K \) and \( r \geq 0 \).

Suppose that the length of the service time is \( x \) and \( j \) primary customers arrive at epochs \( x_1, \ldots, x_j \). This implies
that they may perform retrials during the remaining service time of length \( x = x_1, \ldots, x - x_j \), respectively. It seems very hard to manage these multidimensional constraints. Alternatively, it will be assumed that the \( j \) customers arrive at the mean point \( x/2 \). This simple approximating assumption affects on average only \( \rho>1 \) customers per service. As a result of this reallocation, the estimators \( \hat{z}_{\rho}^{j,k} \) and \( \hat{d}_{\rho}^{j,k} \) are obtained. Moreover, for each fixed \( \rho \geq 0 \), the probabilities \( x_{\rho}^{j}(r) \) satisfy a block tridiagonal system that can be solved iteratively (Amador and Artalejo 2008).

6. NUMERICAL RESULTS AND CONCLUSIONS

Some numerical experiments are presented. It’s considered Erlang-3(\( E_3 \)), exponential (\( M \)) and hyperexponential (\( H_2 \)) service times, with \( \beta_i = 1 \) in all cases. The traffic intensity \( \rho \) takes values 0.2, 0.4, 0.6 and 0.8. The values 0.05, 0.5, 2.5, 25.0 and 100.0 are considered for the retrial rate \( \mu \).

The following results are obtained: \( E[R^s] \) is an increasing function of \( \rho \) but it decreases as function of \( \mu \), and \( E[R^s(H_2)] < E[R^s(M)] < E[R^s(E_3)] \). The behaviour of \( E[A^s] \) is similar to \( E[R^s] \) (Amador and Artalejo 2008). However, the analysis of \( E[R^b] \) is more complicated. In Table 1, it’s showed that \( E[R^b] \) is also increasing with \( \rho \) but it exhibits a minimum as function of \( \mu \). The relationships among the expectations \( E[R^b(H_2)] \), \( E[R^b(M)] \) and \( E[R^b(E_3)] \) depend on the choice of the pair \((\rho, \mu)\).

### Table 1: Value of \( E[R^b] \)

<table>
<thead>
<tr>
<th>( \mu )</th>
<th>0.05</th>
<th>0.5</th>
<th>2.5</th>
<th>25.0</th>
<th>100.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho=0.2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( E_3 )</td>
<td>0.18369</td>
<td>0.18279</td>
<td>0.59421</td>
<td>5.28058</td>
<td>20.9054</td>
</tr>
<tr>
<td>( M )</td>
<td>0.19073</td>
<td>0.23917</td>
<td>0.85894</td>
<td>7.88907</td>
<td>31.3264</td>
</tr>
<tr>
<td>( H_2 )</td>
<td>0.19713</td>
<td>0.28659</td>
<td>1.08216</td>
<td>10.0898</td>
<td>40.1190</td>
</tr>
<tr>
<td>( \rho=0.4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( E_3 )</td>
<td>38.9727</td>
<td>1.26437</td>
<td>2.50721</td>
<td>19.1303</td>
<td>74.6824</td>
</tr>
<tr>
<td>( M )</td>
<td>29.7687</td>
<td>1.50480</td>
<td>3.49664</td>
<td>28.4538</td>
<td>111.783</td>
</tr>
<tr>
<td>( H_2 )</td>
<td>25.5989</td>
<td>1.70860</td>
<td>4.32867</td>
<td>36.3172</td>
<td>143.084</td>
</tr>
<tr>
<td>( \rho=0.6 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( E_3 )</td>
<td>549366.3</td>
<td>12.0361</td>
<td>10.8818</td>
<td>66.3694</td>
<td>253.821</td>
</tr>
<tr>
<td>( M )</td>
<td>145286.3</td>
<td>12.3865</td>
<td>14.4843</td>
<td>98.1345</td>
<td>379.329</td>
</tr>
<tr>
<td>( H_2 )</td>
<td>73420.0</td>
<td>13.0205</td>
<td>17.5211</td>
<td>124.911</td>
<td>485.209</td>
</tr>
<tr>
<td>( \rho=0.8 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( E_3 )</td>
<td>3.3030x10^4</td>
<td>484.278</td>
<td>91.0084</td>
<td>371.393</td>
<td>1370.14</td>
</tr>
<tr>
<td>( M )</td>
<td>2.5939x10^4</td>
<td>341.448</td>
<td>110.462</td>
<td>543.271</td>
<td>2042.12</td>
</tr>
<tr>
<td>( H_2 )</td>
<td>2.7330x10^4</td>
<td>301.639</td>
<td>128.059</td>
<td>688.201</td>
<td>2608.95</td>
</tr>
</tbody>
</table>

The mass probability function of \( R^\rho \) and the mass probability function of \( A^\rho \) are decreasing functions (Amador and Artalejo 2008), but as long as \( \mu \) increases, the mass probability function of \( R^\rho \) may have two modes. One of them is always attached at the initial point \( r=0 \). For \( R^\rho \), Table 2 presents numerical results for \( E_3 \) service times.

It should be noticed that as long as \( \rho \) increases and/or \( \mu \) decreases the distribution of every descriptor under study becomes sparser.

### Table 2: Characteristics of \( R^\rho \)

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( P[R^\rho = 0] )</th>
<th>( P[R^\rho \leq 100] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.92932</td>
<td>0.91914</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.4</td>
<td>0.74966</td>
<td>0.74966</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.6</td>
<td>0.61047</td>
<td>0.67584</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.8</td>
<td>0.51004</td>
<td>0.56643</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Now, the interest is focused on the distribution function of \( R^\rho \). For small values of \( \mu \), the service time tends to expire before the repeated attempt takes place. As a result, the reallocation of customers has no perturbing effect on the system dynamics. Thus, it may be expected high accuracy for small retrial rate. It’s showed that the distribution function of \( R^\rho \) for the reallocation of arrivals gives also a good approximation for higher values of \( \mu \). In addition, the expectations of \( R^\rho \) for the two approximations are compared versus the true value (Amador and Artalejo 2008). The conclusion is that both approximations are very accurate for the three service time distributions.

7. REFERENCES


**AUTHOR BIOGRAPHY**

**JULIA AMADOR** was born in Toledo, Spain and went to the Complutense University of Madrid, where she studied Mathematics. She is Associate Professor in the School of Statistical at the Complutense University of Madrid since 1988. Her research interest is connected to Queueing Systems where she has published several papers. Her e-mail is jamador@estad.ucm.es.
COMPLEX SYSTEMS SIMULATION
A DETERMINISTIC ANNEALING ALGORITHM FOR THE PRE- AND END-HAULAGE OF INTERMODAL CONTAINER TERMINALS

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

KEYWORDS
pickup and delivery, intermodal transportation, meta-heuristics, deterministic annealing

ABSTRACT

The drayage of containers in the service area of an intermodal barge terminal is modelled as a full truckload pickup and delivery problem with time windows (FT-PDPTW). Initial solutions are generated with an insertion heuristic and improved with three local search operators. In a post-optimization phase the three search operators are integrated in a deterministic annealing (DA) framework. The mechanism of the heuristic procedures is demonstrated with a numerical example. A sensitivity analysis indicates that the DA algorithm is robust with respect to variations in threshold value and quality of the initial solution.

INTRODUCTION

Intermodal transport has grown into a dynamic transportation research field. Many new intermodal research projects have emerged. Intermodal transport integrates at least two modes of transport in a single transport chain, without a change of container for the goods, with most of the route traveled by rail, inland waterway or ocean-going vessel and with the shortest possible initial and final journeys by road (Macharis and Bontekoning (2004)). An overview of planning issues in intermodal transport and solution methods proposed in scientific literature is given by Caris et al. (2008). Intermodal planning problems are more complex due to the inclusion of multiple transport modes, multiple decision makers and multiple types of load units. Two strategic planning problems, terminal design and infrastructure network configuration, have received an increased attention in recent years. Yet the number of scientific publications on other intermodal planning problems, especially at the operational decision level, remains limited or non-existent. This paper discusses an operational planning problem in intermodal transport. Pre- and end-haulage of intermodal container terminals involves the pickup or delivery of containers at customer locations. Road transport constitutes a relatively large share of intermodal transport costs. The attractiveness of intermodal transport can be increased by organizing the road segment in the intermodal transport chain more efficiently.

The drayage of containers in the service area of an intermodal terminal may be modelled as a Full Truckload Pickup and Delivery Problem with Time Windows (FTPDPPTW). Savelsbergh and Sol (1995) review the general Pickup and Delivery Problem (PDP). The PDP is an extension to the classical Vehicle Routing Problem (VRP) where customers may both receive and send goods. A fleet of vehicles is required to pickup and/or deliver goods at customer locations. A delivery activity to a consignee starts from the intermodal terminal with a full container and a pickup activity returns a container to the intermodal terminal for shipment by barge. In the Full Truckload Pickup and Delivery Problem (FTPDP) a vehicle carries a single load. In the operational planning problem under investigation, a full truckload is assumed to be a single container. A recent overview of state-of-the-art research on pickup and delivery problems between customers and a depot is presented by Parragh et al. (2008). Only less-than-truckload problems are covered by the authors. Gronalt et al. (2003) study the problem of transporting full truckloads between distribution centres. In their Pickup and Delivery Problem with Time Windows (PDPTW) goods are transported between customer locations, as opposed to our problem definition where all containers either originate or return to the terminal. A full truckload PDPTW is also considered by Currie and Salhi (2003) and Currie and Salhi (2004). The problem studied in these papers also differs from our setting with respect to the definition of customer requests. Goods have to be picked up at works of a construction company and delivered to customers. Wang and Regan (2002) propose a hybrid approach to solve a PDP containing one or more intermodal facilities. Only pickup time windows are considered and the number of vehicles is fixed. The authors apply time window discretization in combination with a
branch and bound method. The closest related article to our research is written by Imai et al. (2007). The authors present a heuristic based on Lagrangian relaxation for the dryage problem of intermodal container terminals, without taking customer time windows into account.

The remainder of this paper is organized as follows. First, the problem formulation is given and a lower bound is proposed. Next, a multi-start local search heuristic is presented to generate an initial solution. This solution is further optimized by a deterministic annealing algorithm. A numerical example demonstrates both heuristic methods. Finally, conclusions are drawn and directions for future research are given.

**PROBLEM FORMULATION**

The FTPDPTW can be formulated in terms of a Vehicle Routing Problem with full container load. Assuming a homogeneous container type and size, the problem is to find an assignment of delivery and pickup customers to a fleet of vehicles, in order to minimize the total cost of serving all customers, which includes fixed vehicle costs and travelling costs. In accordance with Dumas et al. (1991), a fixed vehicle cost is introduced to minimize the fleet size. Each vehicle used incurs a fixed cost, which may vary with the vehicle. Fixed costs include depreciation of own vehicles or leasing costs if the vehicle is hired, insurance payments and fixed costs for hiring an extra truck driver. Travelling costs are proportional to the total time necessary to serve all customers, which implies travelling time and truck waiting time at customer sites. All orders are assumed to be known in advance, so the problem is studied in a static environment. An intermodal terminal is open during a pre-specified daily time window. All trucks have to return to the terminal before the end of their depot window \( (0, T_y) \). Hard time windows at customer locations are assumed.

The FTPDPTW is defined on a graph \( G = (V_0, A) \), where \( V_0 \) represents the node set. \( V \) is the set of all customers, \( V^D \) is the set of delivery customers, \( V^P \) is the set of pickup customers and \{0\} is the singleton representing the depot.

\[
\begin{align*}
V_0 &= V \cup \{0\} \\
V &= V^D \cup V^P \\
V^D \cap V^P &= \emptyset
\end{align*}
\]

The set of arcs \( A \) consists of two types of connections. Arcs either connect the depot with a customer location or provide a connection between two customer locations. Feasible vehicle routes then correspond to paths starting at the depot 0, travelling through arcs connecting customer locations and returning to the depot 0. Only at the beginning and at the end of a route an arc is used to connect a customer location with the depot. The logic of pickup and delivery customers is incorporated in the definition of travel times \( d_{ij} \) of arcs between customer locations. The travel time \( d_{ij} \) of arcs between two customer locations depends on the type of customers served. Four combinations of customers are possible: first a delivery then a pickup customer, two delivery customers consecutively, two pickup customers consecutively or first a pickup and then a delivery customer. Only when a pickup customer is served after a delivery customer a truck can drive directly from one customer location to the other. The travel time \( d_{ij} \) equals the time necessary to move directly from the delivery customer to the pickup customer \( t_{ij} \).

\[
d_{ij} = t_{ij}
\]

In the other three customer combinations the truck first has to return to the depot before serving the second customer. In this case the travel time \( d_{ij} \) is set equal to the time necessary to travel from the first customer to the depot and then from the depot to the second customer.

\[
d_{ij} = t_{i0} + t_{0j}
\]

In this way, the problem can be modelled as a vehicle routing problem with time windows, as described by Cordeau et al. (2007). To formulate the problem the following notation is used:

- \( K = \) set of trucks
- \( x_{ijk} = 1 \) if customer \( i \) and customer \( j \) are served consecutively by truck \( k \), else 0
- \( y_k = 1 \) if truck \( k \) is used, else 0
- \( C_{ijk} \) = travelling cost of arc \((i, j)\) by truck \( k \)
- \( FC_k \) = fixed cost of truck \( k \) for a single route
- \( E_i \) = earliest start time of customer \( i \)
- \( L_i \) = latest start time of customer \( i \)
- \( b_i \) = actual time service at customer \( i \) begins
- \( s_i \) = service time of delivery \( i \)
- \( d_{ij} \) = travel time from customer \( i \) to customer \( j \)
- \( T_k \) = time capacity of truck \( k \)
- \( t_{i0} \) = travel time from terminal 0 to customer \( i \)
- \( t_{ij} \) = travel time directly from delivery \( i \) to pickup \( j \)
- \( t_{i0} \) = travel time from customer \( i \) to terminal 0

\[
\begin{align*}
\min \sum_{i \in V_0} \sum_{j \in V_0, j \neq i} \sum_{k \in K} C_{ijk} x_{ijk} + \sum_{k \in K} FC_k y_k
\end{align*}
\]
\[
\sum_{j \in V_0} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in V \quad (1)
\]

\[
x_{ijk} \leq y_k \quad \forall i, j \in V_0, \quad i \neq j, k \in K \quad (2)
\]

\[
\sum_{i \in V_0} x_{ijk} - \sum_{i \in V_0} x_{jik} = 0 \quad \forall j \in V, k \in K \quad (3)
\]

\[
E_i \leq b_i \leq L_i \quad \forall i \in V \quad (4)
\]

\[
\sum_{k \in K} x_{ijk} \cdot (b_i + s_i + d_{ij} - b_j) \leq 0 \quad \forall i, j \in V \quad (5)
\]

\[
\sum_{k \in K} x_{0jk} \cdot d_{0j} \leq b_j \quad \forall j \in V \quad (6)
\]

\[
x_{rok} \cdot (b_i + s_i + d_{00} - T_k) \leq 0 \quad \forall i \in V, k \in K \quad (7)
\]

\[
\sum_{j \in V} x_{0jk} \leq 1 \quad \forall k \in K \quad (8)
\]

\[
x_{ijk} \in \{0, 1\} \quad \forall i, j \in (V_0), \quad i \neq j, k \in K \quad (9)
\]

\[
y_k \in \{0, 1\} \quad \forall k \in K \quad (10)
\]

\[
b_j \geq 0 \quad \forall i \in V \quad (11)
\]

The objective function minimizes total costs of serving all customers. A fixed vehicle cost \( FC_k \) is incurred for each truck \( k \) used. The variable cost \( C_{ijk} \) represents the cost of serving customer \( j \) immediately after customer \( i \), depending on the travel time and possibly waiting time in case a pickup customer is served directly after a delivery customer. Constraints (1) ensure that each customer is visited exactly once. Constraints (2) avoid to assign customers to unused vehicles. Flow conservation is enforced by constraints (3). Time windows at customer locations are stated in the fourth set of constraints (4). Expressions (5) and (6) enforce the consistency of time variables \( b_j \). Hard time windows are also imposed on the total service time of a route \( k \) by constraints (7). Finally, constraints (8) guarantee that each vehicle is used at most once.

\textbf{LOWER BOUND}

The VRP belongs to the class of NP-hard problems. Since exact models are only able to solve relatively small problems, heuristics are used in practice to solve problems of realistic size. A lower bound is proposed to analyze the performance of the heuristics presented next. According to Cordeau et al. (2007) the LP relaxation of the VRPTW provides a weak lower bound. An alternative formulation is given in this section to be able to calculate a better lower bound for the optimal solution. In this formulation delivery customers are always indicated with index \( i \) and pickup customers with index \( j \). Each route \( k \) consists of a number of trips executed by a single truck \( k \) within the time window \((0, T_k)\). Let a trip be represented as a pair \((i, j)\) where \( i \) represents a delivery customer and \( j \) a pickup customer. Pickup and delivery customers can be combined or can be served separately. In the case only a delivery customer belongs to a trip, the pair is written as \((i, 0)\). If only a pickup customer belongs to the trip, the pair is written as \((0, j)\). In the latter two cases either the delivery point or the pickup point is represented by the depot \( 0 \). This leads to the following alternative notation. All other symbols used, maintain the same interpretation as in the exact problem formulation (formulation (1)-(11)).

\[
V^D_0 = \text{set of delivery points including the depot } 0
\]
\[
V^P_0 = \text{set of pickup points including the depot } 0
\]
\[
x_{ijk} = 1 \quad \text{if delivery } i \text{ and pickup } j \text{ are served in one trip by truck } k, \text{ else } 0
\]
\[
CR_{ijk} = \text{cost of serving pair } (i, j) \text{ by truck } k
\]
\[
RS_{ij} = \text{time necessary to serve pair } (i, j)
\]
\[
E_i = \text{earliest start time of delivery } i
\]
\[
L_i = \text{latest start time of delivery } i
\]
\[
E_j = \text{earliest start time of pickup } j
\]
\[
L_j = \text{latest start time of pickup } j
\]
\[
b_i = \text{actual time delivery } i \text{ begins}
\]
\[
b_j = \text{actual time pickup } j \text{ begins}
\]
\[
l_{0i} = \text{travel time from terminal 0 to delivery } i
\]
\[
l_{ij} = \text{travel time from delivery } i \text{ to pickup } j
\]
\[
l_{0j} = \text{travel time from pickup } j \text{ to terminal 0}
\]
\[
s_i = \text{service time of delivery } i
\]
\[
s_j = \text{service time of pickup } j
\]

\[
\min \sum_{i \in V^D_0} \sum_{j \in V^P_0} \sum_{k \in K} CR_{ijk}x_{ijk} + \sum_{k \in K} FC_k y_k
\]

subject to

\[
\sum_{i \in V^D_0} \sum_{j \in V^P_0} x_{ijk} = 1 \quad \forall j \in V^P \quad (12)
\]

\[
\sum_{j \in V^D_0} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in V^D \quad (13)
\]

\[
x_{ijk} \leq y_k \quad \forall i \in V^D_0, j \in V^P_0, \quad i + j \neq 0, k \in K \quad (14)
\]

\[
E_i \leq b_i \leq L_i \quad \forall i \in V^D \quad (15)
\]

\[
E_j \leq b_j \leq L_j \quad \forall j \in V^P \quad (16)
\]

\[
\sum_{k \in K} x_{ijk} \cdot (b_i + s_i + t_{ij} - b_j) \leq 0 \quad \forall i \in V^D \quad (17)
\]

\[
\sum_{i \in V^D_0} \sum_{j \in V^P_0} RS_{ij} \cdot x_{ijk} \leq T_k \quad \forall k \in K \quad (18)
\]
\[ x_{ijk} \in \{0, 1\} \quad \forall i \in V_q^D, j \in V_q^P, k \in K \]  
(19)

\[ y_k \in \{0, 1\} \quad \forall k \in K \]  
(20)

\[ b_i, b_j \geq 0 \quad \forall i \in V^D, j \in V^P \]  
(21)

In the objective function the variable cost \( CR_{ijk} \) represents the cost of performing the complete trip \((i, j)\) by truck \(k\), including the costs incurred by truck \(k\) to leave and return to the depot. Equations (12) and (13) guarantee that all pickups and deliveries are visited only once. Constraints (14), (15), (16) and (17) are similar to constraints (2), (4) and (5) in the exact formulation. Time windows for the availability of trucks are expressed by constraints (18). The time necessary to perform trip \((i, j)\) is given by the expression:

\[ RS_{ij} = t_{0i} + t_{ij} + t_{j0} + s_i + s_j + MINWAIT_{ij}. \]

The minimum waiting time between delivery customer \(i\) and pickup customer \(j\) equals:

\[ MINWAIT_{ij} = \begin{cases} 0 & \text{if } E_j \leq L_i + s_i + t_{ij} \\ E_j - (L_i + s_i + t_{ij}) & \text{else.} \end{cases} \]

In this formulation the feasibility of the routes is relaxed. If two trips share the same resource (the same vehicle), it is not ensured that the time intervals over which both trips require the resource do not overlap in time. Consequently the lower bound represents the variable costs of optimally combining delivery customers with pickup customers, but underestimates the number of vehicles necessary to perform the selected trips. The lower bound formulation leads to fewer constraints and variables and thus converges more quickly to an integer solution.

**MULTI-START LOCAL SEARCH HEURISTIC**

In the pre- and end-haulage of intermodal containers substantial cost and time savings may be realized by merging pickup and delivery customers in a single trip, as presented in figure 1. A heuristic procedure based on merging pickup and delivery customers is used to construct initial solutions. The insertion heuristic is briefly presented here to better understand the deterministic annealing algorithm. A detailed description and numerical example can be found in Caris and Janssens (2007). Three local search neighbourhoods are defined to improve initial solutions.

**Insertion heuristic**

In this section a two-phase insertion heuristic is described to create initial solutions. In a first phase, pickup and delivery customers are combined into pairs of customers. Due to the existence of hard time windows, not every pickup customer and delivery customer can be combined into a feasible pair. A limit is also imposed on the waiting time between delivery \(i\) and pickup \(j\). This eliminates pairs of customers that are too far away from each other in time. A very large waiting time between the delivery location and pickup location will typically be cost inefficient in road haulage. In forming pairs of pickups and deliveries, both spatial and temporal aspects are to be taken into account. The pairs of pickup and delivery customers are ranked according to four criteria. The time window slack between customers \(i\) and \(j\) should be as small as possible (criterion 1). Savings in travel time obtained from serving delivery \(i\) and pickup \(j\) together should be as large as possible (criterion 2). An opportunity cost for not choosing the best combination for a delivery \(i\) or pickup \(j\) can also be taken into account. Gronalt et al. (2003) argue that this regret approach leads to significant improvements in the best solution. The opportunity cost \( OC_1 \) (respectively \( OC_1^j \)) can be defined as the difference in savings in travel time achieved by the best pair for delivery \(i\) (pickup \(j\)) and the currently selected pair (criterion 3). Finally, the opportunity cost related to the time window slack is incorporated in the selection criterion. This opportunity cost \( OC_2 \) (respectively \( OC_2^j \)) is defined as the difference between the time window slack of the current combination and the smallest time window slack of delivery \(i\) (pickup \(j\)) in any combination (criterion 4). These four criteria are aggregated by making use of weights. The pair of pickup and delivery customers with the lowest value for the following criterion is selected first:

\[
\begin{align*}
&w_1 \cdot (L_j - E_i - s_i - t_{ij}) + w_2 \cdot (t_{ij} - t_{i0} - t_{0j}) \\
&+ w_3 \cdot (OC_1^i + OC_1^j) + w_4 \cdot (OC_2^i + OC_2^j).
\end{align*}
\]

(22)

The weights \(w_1, w_2, w_3\) and \(w_4\) reflect the importance given to each criterion and serve as parameters of the insertion heuristic. The domain of the weights is not fixed. The ratio between the weights influences the importance of each criterion. Depending on the nature of the problem, more weight should be given to savings

![Figure 1: Merging trips](image-url)
in waiting time or savings in travel time. The weights in the insertion heuristic are only used to construct an initial solution, which will be further improved by the local search procedure described below. The process of pairing customers is repeated until no more feasible combinations exist with respect to the remaining pickup customers and delivery customers. The remaining customers are inserted into individual trips and form an imaginary pair with a dummy customer.

In a second phase routes are constructed sequentially. Vehicles are used in increasing order of their fixed costs \( FC_k \). Pairs of customers are eligible to be inserted into routes in increasing order of their latest start time. A pair of customers can be inserted into an existing route \( k \) if it can start later than the time necessary to serve the customers already assigned to the vehicle \( k \) and if the vehicle is able to return to the terminal within its depot window. In case insertion into multiple existing routes is feasible, the pair of customers is added to the existing route with the smallest waiting time between the previous pair. If no insertions into existing routes are feasible, the pair of customers is assigned to an unused vehicle to create a new route.

**Improvement heuristic**

A local search procedure is applied to improve a feasible solution obtained by the insertion heuristic. Three neighborhoods are defined, as presented in figure 2. First, the CROSS operator recombines pairs of customers of different routes. This operator improves the result of the pairing phase in the insertion heuristic. A second operator, COMBINE, joins two routes into one. Finally, customers are removed from a route and inserted into another route by the INSERT operator. The latter two search neighborhoods affect the result of the route construction phase of the insertion heuristic.

![Diagram](image)

**Figure 2: Improvement heuristic**

The CROSS operator selects two pairs of pickup and delivery customers, for example \( (g, h) \) and \( (i, j) \) from two different routes. These pairs are recombined into new pairs of pickup and delivery customers, \( (g, j) \) and \( (i, h) \). First, all feasible CROSS moves are listed. A CROSS move is feasible if the pickup customers and delivery customers can be combined into new pairs, taking into account their time windows. Second, it is checked whether the new pairs of customers can be reinserted into the routes. Either \( (g, j) \) is inserted into the first route and \( (i, h) \) into the second or the other way round. In the local search heuristic the CROSS move with the largest improvement is selected. If a resulting route only contains dummy customers, this route is removed from the solution and the number of trucks necessary is reduced by one. The COMBINE operator checks whether two routes served by different trucks can be combined into a single route. Whereas the first operator reduces the travelling costs in the objective function, the COMBINE operator is able to reduce the number of trucks. Two routes can be combined if the last pair of the first route can be served before the latest starting time of the second route. The third operator, INSERT, removes pairs of pickup and delivery customers from their routes and reinserts them into another route. The INSERT operator attempts to eliminate routes, by inserting their customers into other routes. Pairs of customers can be inserted in the beginning of a route, between pairs of customers or at the end of a route. Similar to the COMBINE operator, this operator also impacts the number of trucks used and consequently the fixed vehicle costs in the objective function. These neighborhood mechanisms are subsets of the general \( \lambda \)-interchange mechanism, described by Osman and Wassan (2002). The CROSS operator is an example of a \( 1 \)-interchange mechanism, which involves only a single customer of each route. Due to the CROSS operator, two routes may exchange either pickup customers or delivery customers of two pairs simultaneously. The INSERT operator represents a \( 2 \)-consecutive-node interchange mechanism. Two consecutive customers which constitute a pair in a single route are shifted to another route. Similarly, the COMBINE operator may be seen a \( n \)-consecutive-node interchange mechanism.

A multistart approach using different values for the weights in selection criterion (22) of the insertion heuristic may be applied to obtain the best overall solution.

**DETERMINISTIC ANNEALING**

A deterministic annealing algorithm is applied in a post-optimization phase to further improve on solutions found by the multistart local search heuristic. Deterministic annealing (DA), also referred to as 'threshold accepting', is introduced by Dueck and Scheuer (1990) as a deterministic variant of simulated annealing (SA). In each step of an SA algorithm a new solution \( S' \) is generated in the neighbourhood of the current solution \( S \). If the new solution has a better objective value, it
is accepted automatically. If it is worse, it is accepted only with a certain probability. The probability of acceptance \( e^{-\Delta/T} \) depends on the change in objective value \( \Delta = C(S') - C(S) \) and a parameter \( T \), called temperature. The temperature \( T \) is updated according to a certain annealing schedule. In the beginning of the search \( T \) is set at a level with a high probability of accepting worse solutions. Gradually, the probability of accepting deteriorations is lowered, until only improvements are accepted. A great variety of annealing schedules exist in literature. However, Dueck and Scheuer (1990) state that in most applications the success of SA is very sensitive against the choice of annealing schedule. Deterministic annealing offers a greater simplicity. The difference between SA and DA lies in the different acceptance rules. In DA a neighbouring solution with a worse objective value than the current solution is accepted if the deterioration \( \Delta = C(S') - C(S) \) is less than a deterministic threshold value \( T \).

Applications of DA to vehicle routing problems can be found amongst others in Tarantilis et al. (2004) and Bräysy et al. (2008). DA is applied to the problem formulation described in this paper based on the implementation strategy of Bräysy et al. (2008). The final solution of the multistart local search heuristic serves as initial solution for the DA algorithm, presented in Algorithm 1.

**Algorithm 1** Deterministic annealing for FTPDPTW

Set best solution of multistart local search heuristic as current best solution \( S_{\text{best}} \) of DA

Set \( T = T_{\text{max}} \) and \( i_{\text{last}} = 0 \)

for \( i = 1 \) to \( n_{\text{improve}} \) do

Choose two random starting routes

for All route pair combinations do

Apply CROSS

Apply COMBINE

Apply INSERT

end for

if \( C(S') < C(S_{\text{best}}) \) then

Set \( S_{\text{best}} = S' \) and \( i_{\text{last}} = i \)

else

if \( T \leq 0 \) and \( i - i_{\text{last}} \geq \bar{n} \) then

Restart from \( S_{\text{best}} \):

Set \( S' = S_{\text{best}} \), \( i_{\text{last}} = i \) and \( T = r \cdot T_{\text{max}} \)

else

if \( T \leq 0 \) then

Set \( T = r \cdot T_{\text{max}} \)

else

Set \( T = T - \Delta T \)

end if

end if

end if

end for

The three local search neighbourhoods CROSS, COMBINE and INSERT are integrated in a deterministic annealing framework. Routes are searched in a fixed order, but at the beginning of each iteration the starting point of the search is randomly chosen. Neighbouring solutions with a worse objective value are accepted when \( \Delta = C(S') - C(S) \) is less than the threshold value \( T \). For each pair of routes at most one move for each local search operator is accepted in each iteration. In the DA algorithm a first-accept strategy is applied, whereas in the local search heuristic in the previous section the best move was chosen. The threshold value is initially set at a maximum value \( T_{\text{max}} \). In each iteration without any improvement in objective function value \( T \) is lowered with \( \Delta T \) units. The threshold value is reset to \( r \cdot T_{\text{max}} \) whenever it reaches zero, with \( r \) a random number between 0 and 1. When after a predefined number of iterations \( \bar{n} \) no improvements have been found and \( T \) reaches 0 again, the algorithm restarts from the current best solution \( S_{\text{best}} \) found. The process is repeated for \( n_{\text{improve}} \) number of iterations.

**NUMERICAL EXAMPLE**

A numerical example is discussed to demonstrate the mechanism of the heuristic procedures. In this example an intermodal terminal has to pickup or deliver containers to a hundred customer sites. The terminal is open during eight hours per day. Service at customer sites takes eight minutes. The problem is studied in a deterministic environment. Travel times, waiting times and service times are therefore assumed to be constant. Customer locations are randomly selected with \( x \) - and \( y \)-coordinates between zero and 25. Time windows at customer locations are randomly chosen between 60 and 120 minutes. The terminal cooperates with a single hauler for performing the road segment of intermodal transport requests. Therefore, travelling costs and fixed vehicle costs are assumed equal for all vehicles. A fixed vehicle cost of 10 is charged per vehicle in use.

In the insertion heuristic a maximum waiting time between delivery customers and pickup customers of 30 minutes is allowed. A multistart approach is applied, varying the weights in selection criterion (22). The weights are altered from zero to 100 with increases of five units. The weights always sum up to 100. The best overall solution is obtained with the weights reported in table 1. A large weight is given to the opportunity costs of savings in travel time. No weight is allocated to the time window slack between customers or opportunity costs of time window slack.

Table 2 presents the variable cost (VC) or travelling cost, fixed vehicle cost (FC) and total cost (TC) of the best overall solution after applying the insertion heuristic and the three local search neighbourhoods. The insertion heuristic serves to provide an initial solution.
Table 1: Weights best overall solution multistart local search heuristic

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>80</td>
<td>0</td>
</tr>
</tbody>
</table>

This initial solution is strongly improved by the three local search operators. The CROSS operator reduces the variable cost, whereas the two other operators are aimed to decrease the fixed vehicle cost. The final total cost differs only 1.94% from the lower bound. In the lower bound solution less vehicles are required, due to the relaxation of route feasibility with respect to the customer time windows.

<table>
<thead>
<tr>
<th></th>
<th>VC</th>
<th>FC</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insertion heuristic</td>
<td>2903</td>
<td>90</td>
<td>2993</td>
</tr>
<tr>
<td>CROSS</td>
<td>2768</td>
<td>90</td>
<td>2858</td>
</tr>
<tr>
<td>COMBINE</td>
<td>2768</td>
<td>90</td>
<td>2858</td>
</tr>
<tr>
<td>INSERT</td>
<td>2768</td>
<td>70</td>
<td>2838</td>
</tr>
<tr>
<td>Lower bound</td>
<td>2734</td>
<td>50</td>
<td>2784</td>
</tr>
</tbody>
</table>

Table 2: Multistart local search heuristic

Deterministic annealing is applied as a post-optimizer to further reduce the total cost of this solution. In the DA algorithm the number of iterations $n_{\text{improve}}$ is fixed at 200. The algorithm is restarted from the current best solution $S_{\text{best}}$ after 10 iterations without any improvements $\bar{n}$ with the threshold value at zero. The maximum threshold value $T_{\text{max}}$ equals two, with a change in threshold value $\Delta T$ of 0.025. Results of three independent runs of the DA algorithm are given in table 3. The DA algorithm finds further reductions in travelling costs. The three runs show similar results with a gap of around 1% between the heuristic solution and the lower bound.

<table>
<thead>
<tr>
<th></th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC</td>
<td>2737</td>
<td>2740</td>
<td>2748</td>
</tr>
<tr>
<td>FC</td>
<td>70</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>TC</td>
<td>2807</td>
<td>2810</td>
<td>2818</td>
</tr>
<tr>
<td>Gap</td>
<td>0.82%</td>
<td>0.93%</td>
<td>1.22%</td>
</tr>
</tbody>
</table>

Table 3: Deterministic annealing algorithm

Figure 3 presents a sensitivity analysis to determine the maximum threshold value $T_{\text{max}}$. The maximum threshold value is varied from zero to five with intervals of size 0.2. Three independent test runs are performed for each value. The vertical axis shows the percentage deviation of the median objective function value with respect to the lowest total cost over all threshold values and test runs. A maximum threshold value of at least 1.4 is appropriate to obtain the lowest percentage deviation. Maximum threshold values less than one do not allow enough diversity in the search process. Only minor deviations are found over all threshold values, which indicates that the DA algorithm is robust for changes in maximum threshold value.

In figure 4 the influence of parameter $\Delta T$ on the solution quality is investigated in a similar way. The maximum threshold value is held constant at $T_{\text{max}} = 2$. Only small deviations from the lowest objective function value are measured, showing the robustness of the DA algorithm for changes in $\Delta T$.

In table 4 multiple initial solutions are tested for the deterministic annealing algorithm. Ten different initial solutions are generated by assigning the values in columns one to four to the weights in selection criterion (22). For each initial solution three independent test runs are performed. The fifth column (TC) mentions the median objective function value. The percentage deviation
from the minimum value is reported in column six (% dev). The total costs differ only slightly. A comparison of table 4 with table 3 shows that the lowest overall cost results from the best solution generated by the multi-start local search heuristic.

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>TC</th>
<th>% dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>2845</td>
<td>0.23</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>2838</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2842</td>
<td>0.14</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2839</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2839</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>2850</td>
<td>0.42</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2845</td>
<td>0.23</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2838</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2850</td>
<td>0.42</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2849</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 4: Sensitivity analysis of initial solution

CONCLUSIONS AND FUTURE WORK

In this paper a deterministic annealing algorithm is presented to find near optimal solutions for the drayage of containers in the service area of intermodal terminals. The DA algorithm is based on three local search operators, CROSS, COMBINE and INSERT. A preliminary analysis is performed with a numerical example. The DA algorithm generates good quality solutions independent of the quality of the initial solution. In the future computational experiments will be set up to confirm the robustness of the algorithm with respect to variations in problem characteristics.

REFERENCES


MODELLING AND SIMULATION OF THE PHARMACY FOR THE UNIVERSITY HOSPITAL CENTER OF ORAN (ALGERIA)

Khaled Belkadi
LAMOSI, University of Mohamed Boudiaf, USTO, BP 1505 Oran M’Naouer, 31000 Oran, Algeria
E-mail: belkadi1999@yahoo.com, belkadi@isima.fr

Alain Tanguy
LIMOS, CNRS UMR 6158, University Blaise Pascal, Les Cezeaux, 63173 Aubiere Cedex, France
E-mail: tanguy@isima.fr

KEYWORDS
Hospital, Pharmacy service, ARIS modelling, SIMULA model, Discrete-event simulation.

ABSTRACT

Pharmacy is an essential part of the hospital. Its mission is the supply and delivery of pharmaceutical products to various hospital services. These products are mainly drugs, dressings, instrumentations... The aim of this paper is to model and simulate the pharmacy of University Hospital Center of Oran (UHCO) in Algeria using ASCI methodology. The pharmacy service is mainly composed of two Sub-services: Unloading and Distribution. Unloading is to unload, control and seize products. The distribution is to control, seize and distribute products. The objective is to study the processing time for each unit or section of the unloading and distribution, and processing time for each person connected to the pharmacy service. This will enable us to improve performance. To achieve this goal, we first used the ARIS tool to specify the knowledge model and then the simulation thanks to the SIMULA language to implement action models.

INTRODUCTION

Hospital systems are complex systems in which problems have to be solved such as their size, the enhancement of their efficiency or obviously the understanding of their operation. These problems concern performance evaluation they can be solved using modelling and simulation.

Modelling is a decision aid tool which prevents important financial investments. This paper describes the modelling and the simulation of the pharmacy service in the hospital system. The ASCI (Analysis, Specification, Conception and Implementation) (Gourgand and Kellert 1991) modelling methodology is adapted and used for this system. It is based on the construction of two models classes: the knowledge model and the action models.

The aim of this work is to model and to simulate the pharmacy service of the University Hospital Center of Oran (UHCO), following ASCI methodology, using ARIS tool (Architecture of Integrated Information System) (Sheer 2002) for the knowledge model and the SIMULA language for the action models.

The pharmacy service of Oran UHCO is composed mainly of two Sub-services: Unloading and Distribution. Both Sub-services handle pharmaceuticals products (drugs, dressings, instrumentation...). For each Sub-service, we have seven units or sections: Unit Drug, Unit Mass Solutes, Section

Single use, Section Dressings, Section Instrumentation, Section Consumable Product and Section Laboratory. Each unit or section corresponds to products (drugs, dressings, instrumentations...). For the unloading of trucks, suppliers come with the products for each unit or section. These products are unloaded, controlled and seized by qualified persons before being distributed. For distribution, the products are controlled, seized and distributed to concerned services or functional units (Central Radiology, Epidemiology, Medical Emergency, Immunology laboratory ...).

Our goal is to study the processing time per each unit or section and per every person incorporated in the pharmacy service and especially at the unloading and distribution level. That enables us to estimate the total work time per day or a longer duration.

PHARMACY SERVICE OF THE UHCO

The UHCO in Oran incorporates several administrations. The different services of the UHCO are located in the UHCO (Chair) and in the different clinics that are attached to it. The UHCO surface is 13 hectares. It is composed of 42 services with 9 extra muros. It has a total capacity of more than 1488 beds with 700 beds dedicated to general medicine, 541 beds for surgery, 136 beds for gynaecology and 111 beds for paediatrics. It employs 6000 people with 1312 medical employees, 1751 paramedical employees, 1761 administrative employees and the remaining combines: handymen, janitors and security employees. Pharmacy is one of the busiest services in the hospital. Its mission is the supply and delivery of pharmaceutical products to hospital services. Pharmacy is composed of different services and is used by several services (Barka et al. 2007). The providers of the “Central Pharmacy of UHCO” belong to one category: Public or Private.

MODELLING METHODOLOGY

The modelling methodology ASCI has been adapted to hospital systems (Gourgand and Combes 1994; Mebrek and Tanguy 2006; Mebrek, et al. 2007). The knowledge model describes the structure and the operating principle of the system in a natural or graphical language; it is built thanks to three subsystems (logical, physical and decisional). An action model is a translation of the knowledge model in a mathematical formalism or in a programming language enabling the evaluation of chosen performance criteria.

The main goal of the modelling methodology is to establish a knowledge model that is as generic as possible and that allows the execution of the action models specific to the
systems of the domain. The knowledge model remains an open model which is enhanced by each domain systems study. The management of the knowledge and the execution of the action models imply the help of an open modelling environment in order to include new and more efficient methods and tools. The modelling environment (figure 1) cases information exchange between the project members and helps the conception of action models during the extraction of the information from the knowledge model. It's an attempt to introduce automatism in the modelling process with the formalisation of the knowledge, the analysis of the data to determine the characteristics of the system, the operational research and the simulation for the evaluation. Graphical representations and animation tools help verifying proper operation of the model. The first knowledge model of the hospital logical system has been formalised thanks to the ARIS tool that is appropriate to describe organisations, processes and activities (Green and Roseman 2000), as well as entity-relationship models (Chen 1976).

The knowledge model is a formalised description of the system that contains the acquired knowledge during the observation phase of the existing system or the specification of the topology and functioning stated by the designers. The action model is a translation of the knowledge model using a mathematical formalism (for example an analytical method which takes advantage of the queuing network theory) or in a programming language (for example a simulation language). It is directly usable and states the performances of the modelled system without using direct measure. Exploitation of the knowledge model and of the action model is called modelling process. This process is generally iterative and consists in four steps which are the elaboration of a system knowledge model, the translation of this knowledge model into an action model, the exploitation of the action model to evaluate the performances of the system and the interpretation of the results and consequently to deduce the modifications to be made on the system. Each step includes a verification and validation phase. A knowledge model has a wide application area.

In order to use ARIS to design a knowledge model, several modelling hypothesis are to be taken into account:
- Each activity (function in ARIS) is linked to one or more organisational units of the hospital system (care unit, operating room, the pharmacy, etc.);
- Each event possess its own information document, it is used by several processes and it is referenced in one or more documents of the information system (medical file of the patient, file of the operating room suite, etc.);
- The referenced documents provide the knowledge concerning the key processes.

To match our modelling goals, we chose the ARIS tool-set and we retain two representation types:
- The event-driven process chain (EPC) in order to show that the processes have a well defined structure and to control the logical subsystems flows;
- The organisational structure for the decisional subsystem to detail the relationships in and between the services.

**KNOWLEDGE MODEL EPC**

The sequence of functions in the sense of an enterprise process is represented in process chain. In these chains, it is possible to indicate the departure and arrival events for each function. The events trigger functions and they are generated by them. Event-driven process chain (EPC) represents the organisational structure of the enterprise, i.e. the representation of the relationships between the data view, objects, the functions and the organisational views. An EPC describes the sequencing of functions. For each function, an initial event and a final event are defined. The events trigger the functions and a function generates events. Function and event are represented by a rounded rectangle and a hexagon (figure 3).
As the events define the state or the condition that triggers a function as well as the end state, the start and end nodes of an EPC are always events. An event can trigger several functions simultaneously and a function can provoke several events. To represent the links and the processing loops of an EPC, the system uses a connector (or ruler) which as the shape of a circle. Figure 4 shows the specification of the pharmacy service by an EPC, starting at truck arrival and ending by distribution to the last functional unit (UF).

**Figure 4: Pharmacy EPC**

**ACTION MODELS**

The pharmacy service action (or simulation) model of the “University hospital center of Oran” is represented by the SIMULA model.

**Queuing network model**

The figures 5 and 6 show how to use the waiting queue model to represent an action (or simulation) model of the unloading and distribution in the Pharmacy service. For unloading and distribution, we have seven units or sections: the Unit Drug, the Unit Mass Solutes, Section Single use, Section Dressings, Section Instrumentation, Section Consumable Product and Section Laboratory. Each unit or section corresponds to products (drugs, dressings, instrumentations,…). For the unloading of trucks (suppliers) come with the products for each unit or section. These products are unloaded, controlled and seized by qualified persons before being distributed.

**Figure 5: Queuing network model of the unloading**

For distribution, the products are controlled, seized and distributed to concerned services or functional units (Central Radiology, Epidemiology, Medical Emergency, Immunology laboratory, …).

**Figure 6: Queuing network model of the distribution**

**SIMULA model**

The SIMULA language proved its capacity to implement different simulation models categories. It includes
coroutines and processes of discrete events simulation. Numerous classes exist that extend the language possibilities regarding transactions management and statistical computations. The Gpss class provides base objects such as the service, the storage, the transaction notions as well as the statistical region. Moreover, a simulation report is generated automatically. This class can therefore be used in GPSS programming with all the object-oriented capacity of a simulation language.

RESULTS AND INTERPRETATIONS

We obtained results with data taken on a real case for the Unloading and the Distribution of the Pharmacy service of the UHCO in Oran (Barka et al. 2007). The default time unit is minute.

Table 1 contains the data of model by unit or section for unloading. These values represent actual data taken at the UHCO pharmacy service for unloading.

<table>
<thead>
<tr>
<th>Min-Max</th>
<th>Person</th>
<th>Unloading</th>
<th>Seizure</th>
<th>Control</th>
<th>Supply</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Drugs</td>
<td>2-5</td>
<td>45-60</td>
<td>60-120</td>
<td>60-120</td>
<td>1-3</td>
<td>1/month</td>
</tr>
<tr>
<td>Unit Mass Solutes</td>
<td>1-2</td>
<td>30-60</td>
<td>60-120</td>
<td>30-60</td>
<td>1-3</td>
<td>1/month</td>
</tr>
<tr>
<td>Section single-use</td>
<td>2-5</td>
<td>30-60</td>
<td>60-120</td>
<td>60-120</td>
<td>1-3</td>
<td>1/month</td>
</tr>
<tr>
<td>Section dressings</td>
<td>3-5</td>
<td>30-60</td>
<td>60-120</td>
<td>60-120</td>
<td>1-3</td>
<td>1/7 days</td>
</tr>
<tr>
<td>Section Instrument</td>
<td>1-2</td>
<td>30-60</td>
<td>480-960</td>
<td>60-120</td>
<td>1-3</td>
<td>1/3 months</td>
</tr>
<tr>
<td>Section consumable</td>
<td>2-5</td>
<td>30-45</td>
<td>30-60</td>
<td>30-60</td>
<td>1-3</td>
<td>1/month</td>
</tr>
<tr>
<td>Laboratory</td>
<td>2-5</td>
<td>30-45</td>
<td>30-60</td>
<td>30-60</td>
<td>1-3</td>
<td>1/month</td>
</tr>
</tbody>
</table>

Table 2 contains the data of model for distribution. These values represent actual data taken at the UHCO pharmacy service for distribution.

<table>
<thead>
<tr>
<th>Seizure Min-Max</th>
<th>Control Min-Max</th>
<th>Service (UF) Min-Max</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Drugs</td>
<td>10-15</td>
<td>5-10</td>
<td>8-12</td>
</tr>
<tr>
<td>Unit Mass Solutes</td>
<td>10-15</td>
<td>5-10</td>
<td>8-12</td>
</tr>
<tr>
<td>Section single-use</td>
<td>10-15</td>
<td>5-10</td>
<td>3-7</td>
</tr>
<tr>
<td>Section dressings</td>
<td>10-15</td>
<td>5-10</td>
<td>3-7</td>
</tr>
<tr>
<td>Section Instrumentations</td>
<td>10-15</td>
<td>5-10</td>
<td>3-7</td>
</tr>
<tr>
<td>Section consumable product</td>
<td>10-15</td>
<td>5-10</td>
<td>3-7</td>
</tr>
<tr>
<td>Laboratory</td>
<td>10-15</td>
<td>5-10</td>
<td>3-7</td>
</tr>
</tbody>
</table>

The obtained results with the SIMULA language and Montreal Gpss class for Unloading are given in figure 7. The simulated time is 400 minutes.

* facilities *

<table>
<thead>
<tr>
<th>Entries</th>
<th>Average contents</th>
<th>Average time</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Drugs</td>
<td>2</td>
<td>0.99</td>
<td>236.34</td>
</tr>
<tr>
<td>Unit Mass Solutes</td>
<td>3</td>
<td>0.98</td>
<td>157.11</td>
</tr>
<tr>
<td>Section single-use</td>
<td>2</td>
<td>0.98</td>
<td>236.10</td>
</tr>
<tr>
<td>Section dressings</td>
<td>2</td>
<td>0.98</td>
<td>236.11</td>
</tr>
<tr>
<td>Section Instrument</td>
<td>1</td>
<td>0.98</td>
<td>471.41</td>
</tr>
<tr>
<td>Section consumable</td>
<td>3</td>
<td>0.84</td>
<td>134.21</td>
</tr>
<tr>
<td>Section Laboratory</td>
<td>2</td>
<td>0.58</td>
<td>139.09</td>
</tr>
</tbody>
</table>

* storages *

<table>
<thead>
<tr>
<th>Entries</th>
<th>Average contents</th>
<th>Average time</th>
<th>Cont. now</th>
<th>Cont. max</th>
<th>Capacity</th>
<th>Util. %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Drugs</td>
<td>2</td>
<td>0.67</td>
<td>55.26</td>
<td>0</td>
<td>3</td>
<td>2.21</td>
</tr>
<tr>
<td>Control Drugs</td>
<td>2</td>
<td>0.43</td>
<td>102.60</td>
<td>0</td>
<td>1</td>
<td>42.75</td>
</tr>
<tr>
<td>Seizure Drugs</td>
<td>2</td>
<td>0.30</td>
<td>71.00</td>
<td>0</td>
<td>1</td>
<td>29.28</td>
</tr>
<tr>
<td>PersonMassSolutes</td>
<td>3</td>
<td>0.28</td>
<td>45.04</td>
<td>0</td>
<td>1</td>
<td>28.15</td>
</tr>
<tr>
<td>ControlMassSolutes</td>
<td>3</td>
<td>0.21</td>
<td>34.12</td>
<td>0</td>
<td>1</td>
<td>21.32</td>
</tr>
<tr>
<td>SeizureMassSolutes</td>
<td>3</td>
<td>0.44</td>
<td>70.46</td>
<td>0</td>
<td>1</td>
<td>44.04</td>
</tr>
<tr>
<td>PersonSingleUse</td>
<td>2</td>
<td>0.38</td>
<td>45.38</td>
<td>0</td>
<td>2</td>
<td>18.99</td>
</tr>
<tr>
<td>ControlSingleUse</td>
<td>2</td>
<td>0.42</td>
<td>101.01</td>
<td>0</td>
<td>1</td>
<td>42.42</td>
</tr>
<tr>
<td>SeizureSingleUse</td>
<td>2</td>
<td>0.34</td>
<td>81.69</td>
<td>0</td>
<td>1</td>
<td>34.04</td>
</tr>
<tr>
<td>PersonDressings</td>
<td>2</td>
<td>0.88</td>
<td>52.09</td>
<td>0</td>
<td>4</td>
<td>22.04</td>
</tr>
<tr>
<td>ControlDressings</td>
<td>2</td>
<td>0.40</td>
<td>94.00</td>
<td>0</td>
<td>1</td>
<td>39.90</td>
</tr>
<tr>
<td>SeizureDressings</td>
<td>2</td>
<td>0.34</td>
<td>81.70</td>
<td>0</td>
<td>1</td>
<td>34.04</td>
</tr>
<tr>
<td>PersonInstrument</td>
<td>1</td>
<td>0.08</td>
<td>37.74</td>
<td>0</td>
<td>1</td>
<td>7.86</td>
</tr>
<tr>
<td>ControlInstrument</td>
<td>1</td>
<td>0.16</td>
<td>75.94</td>
<td>0</td>
<td>1</td>
<td>15.82</td>
</tr>
<tr>
<td>SeizureInstrument</td>
<td>1</td>
<td>0.73</td>
<td>349.00</td>
<td>0</td>
<td>1</td>
<td>72.73</td>
</tr>
<tr>
<td>PersonConsumable</td>
<td>3</td>
<td>0.93</td>
<td>37.28</td>
<td>0</td>
<td>4</td>
<td>23.30</td>
</tr>
<tr>
<td>ControlConsumable</td>
<td>3</td>
<td>0.27</td>
<td>42.39</td>
<td>0</td>
<td>1</td>
<td>26.49</td>
</tr>
<tr>
<td>SeizureConsumable</td>
<td>3</td>
<td>0.30</td>
<td>47.60</td>
<td>0</td>
<td>1</td>
<td>29.75</td>
</tr>
<tr>
<td>PersonLaboratory</td>
<td>2</td>
<td>0.35</td>
<td>41.44</td>
<td>0</td>
<td>2</td>
<td>17.27</td>
</tr>
<tr>
<td>ControlLaboratory</td>
<td>2</td>
<td>0.19</td>
<td>46.15</td>
<td>0</td>
<td>1</td>
<td>19.23</td>
</tr>
<tr>
<td>SeizureLaboratory</td>
<td>2</td>
<td>0.19</td>
<td>44.35</td>
<td>0</td>
<td>1</td>
<td>18.48</td>
</tr>
</tbody>
</table>

* regions *

<table>
<thead>
<tr>
<th>Entries</th>
<th>Average contents</th>
<th>Average time</th>
<th>Cont. now</th>
<th>Cont. max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Drugs</td>
<td>2</td>
<td>0.99</td>
<td>236.34</td>
<td>1</td>
</tr>
<tr>
<td>Mission Mass Solutes</td>
<td>3</td>
<td>0.98</td>
<td>157.11</td>
<td>1</td>
</tr>
<tr>
<td>Mission Single Use</td>
<td>2</td>
<td>0.98</td>
<td>236.10</td>
<td>1</td>
</tr>
<tr>
<td>Mission Dressings</td>
<td>2</td>
<td>0.98</td>
<td>236.11</td>
<td>1</td>
</tr>
<tr>
<td>Mission Instrument</td>
<td>1</td>
<td>0.98</td>
<td>471.41</td>
<td>1</td>
</tr>
<tr>
<td>Mission Consumable</td>
<td>3</td>
<td>0.84</td>
<td>134.21</td>
<td>0</td>
</tr>
<tr>
<td>Mission Laboratory</td>
<td>2</td>
<td>0.58</td>
<td>139.09</td>
<td>0</td>
</tr>
</tbody>
</table>

* tables *

<table>
<thead>
<tr>
<th>Unit or section name</th>
<th>Unloading</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drugs</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Mass Solutes</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Single Use</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Dressings</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Instrumentations</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Consumable product</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Laboratory</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Total Unloading number</td>
<td>17</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: SIMULA model results for Unloading

For the Unloading, the section workload is included between 58% and 99% of the simulated time. The occupancy rate of the unloading employees is included between 8% and 28%, the one of the control between 16% and 43% and the one for data capture between 19% et 73%. It is only a 8 hours simulation. The average data capture time is more important than the unload and control time. It is due to too long data capture slots between 30 and 960 minutes which lead to the saturation of the service when the planning is full. The test was obviously conducted in extreme functioning conditions. For this it is necessary to add persons at this level and especially for the instrumentation section.

The obtained results with the SIMULA model for the Distribution are given in figure 8. The simulated time is (7*8*60) = 3360 minutes.
stay is very important in relation to the duration of other sections. This is due to the significant number of arrived products which is at 8 and 12 products. But we will not forget that for these two units, the distribution is weekly and why it is necessary to spend more than 70 products during the 7 days period.

CONCLUSION

We presented the pharmacy service of the UHCO and the usefulness of the knowledge model specified thanks to the ARIS tool, as well as the transition from this model to the action (or simulation) model implemented with the SIMULA language. Most works (articles) on pharmacies focus on inventory management or monitoring of medicines for patients staying in hospitals. Our contribution in this paper is the study of two services (unloading and distribution) at the UHCO pharmacy which generally are not taken into account in studies. The unloading is an upstream work of pharmacy which is to unload, control and seize products. The distribution is a downstream work of pharmacy which is to control, seize and distribute products. The main problem is the study of the processing time for each unit or section and person attached to the pharmacy service and the total time after a working day or more. According to the results, we note, for example, that the seizure phase of information on products takes a long time. To remedy this, we suggest for example, adding additional personnel, or replace the seized agents by a bar code reader connected to the computer for example. Several research themes can complete this work: modelling and simulate other hospital services, using other modelling and simulation tools like Witness and Flexsim, studying the planning of hospital services and the driving of the hospital systems.

REFERENCES


Figure 8: SIMULA model results for Distribution

For the Distribution, the work of each unit or section is included between 27% and 62% of the simulated time. The occupancy rate of the distribution employees is included between 19% and 45%, the one for data capture between 12% and 28%.

It is only a 7*8 hours (7 working days) simulation. Note that for unit Drugs and Mass solutes, the average time of
COOPERATING MODELLING METHODS
FOR PERFORMANCE EVALUATION OF INTERCONNECTED
INFOCOMMUNICATION AND BUSINESS PROCESS SYSTEMS

László Muka
Ellassys Consulting Ltd.
Bég utca 3-5.
H-1026 Budapest, Hungary
muka.laszlo@ellassys.hu

Gábor Lencse
Department of Telecommunications
Széchenyi István University
Átjáró tér 1.
H-9026 Győr, Hungary
lencse@sze.hu

KEYWORDS
discrete-event simulation, traffic-flow analysis, entity flow phase analysis, information and communication systems, business process systems, efficiency of simulation, system-model matrices, preliminary modelling, translation of conceptual models, rapid models

ABSTRACT
This paper describes how the rapid and detailed modelling and simulation methods can be used to increase the efficiency of mixed simulation projects initiated to support the design of interconnected ICT (Information and Communication Technology) and BP (Business Process) systems. A system of cooperating rapid and detailed methods for critical and non-critical parts of systems is introduced in the phase of preliminary and detailed modelling. The method of translation of information of conceptual models that had been built prior to simulation is described, too. New methods using rapid models to model the ICT and BP systems functioning as environment for the detailed models are presented. A novel method for preliminary modelling based only on cooperating system of rapid models is described.

INTRODUCTION
Simulation projects aimed at supporting the design of Information and Communication Technology (ICT) systems and Business Process (BP) systems in an organisation are usually separate projects but these systems may have significant influence on each other, therefore common analysis of these systems in mixed simulation projects may have advantages. In mixed simulation projects we need to have methods appropriate for both types of systems: we need models of ICT and BP systems that can interact with each other just as these systems interact with each other in the real world.

The efficiency of the modelling and simulation process is influenced by two main components: the tool dependent and the tool independent component. The tool dependent component means the simulator and its services (how the models can be built, how the experimenting and debugging is supported, etc.). This approach is well introduced in (vom Lehn at al. 2008). By tool independent component we mean the set of modelling and simulation methods, as they can be implemented in any simulation environment. In this paper we will focus on some of these methods.

Discrete-Event Simulation (DES) can be used for detailed and accurate analysis and performance evaluation of ICT systems (Jain 1991) and BP systems. Simulation models of ICT systems (DES-IT for short in this paper) and simulation models of BP systems (DES-P for short in this paper) have similarities but their semantics are different.

The fast and approximate performance estimation can be very useful in the early stage of simulation projects. The Traffic-Flow Analysis (TFA), which is a combination of simulation and statistical approaches (Lencse 2004), was proposed for the rapid modelling and preliminary performance estimation of ICT systems.

The Entity Flow-Phase Analysis (EFA), which was derived from TFA, (Lencse and Muka 2006) is a method for the rapid modelling and preliminary investigation of BP systems.

In the case of large and complex systems, with which we may be easily faced in mixed simulation projects, the necessary computing capacity may reach the power of a supercomputer. Traditional parallel simulation methods (Parallel Discrete Event Simulation, PDES) (e.g., conservative, optimistic) (Fujimoto 1990) can rarely provide an attractive speed-up (Pongor 1992).

In mixed simulation projects, the combination of cooperating detailed (DES-IT, DES-P) and rapid (TFA, EFA) models and methods based on the outputs of Modified Conceptual Modelling (MCM), may give the required increase of efficiency. (Modified Conceptual Modelling (Muka and Lencse 2006), which is a method of merging hard-systems and soft-systems approaches, is a powerful model design support tool in simulation projects.)

SYSTEM-MODEL MATRICES

System-Model Matrix of Cooperation for Detailed Modelling

In the system-model matrix for detailed modelling (Figure 1) there are four types of models: DES-IT and DES-P models for modelling the critical ICT and BP systems (or critical parts of systems) respectively, and we have TFA and EFA models for modelling the non-critical ICT and BP systems (or non-critical parts of systems). We also show the cooperation between systems (denoted by numbers 1-6 in Figure 1).
In the case of the analysis of interconnected ICT and BP systems, we may have three basic situations:

1. Both ICT and BP systems are in the focus of the analysis. In this case the ICT system is modelled by a DES-IT model and the BP system by a DES-P model. Cooperating DES-IT and DES-P models have to use change of interpretations in their communication (number 2 in Figure 1). If the ICT system has a non-critical part a TFA model can be used for modelling it. Between cooperating DES-IT and TFA models there is a change of representations in communication (number 1 in Figure 1). In case the BP system, the EFA model is used to model the non-critical part and there is also a change of representations in the communication between EFA models and DES-P models (number 3 in Figure 1). If there are both TFA and EFA models in the model of the whole system then all the six types of communications can be used between them (number 1-6 in Figure 1) with the necessary changes of interpretations and representations.

2. The ICT system is in the focus of the examination and the BP system is the environment for the ICT system. In this case the ICT system is modelled by a DES-IT model and the BP environment is modelled by an EFA model. The communication between cooperating DES-IT and EFA models is denoted by number 6 in Figure 1. (The question of change of representation and interpretation should be answered in this case, too.)

3. The BP system is in the focus of the examination and the ICT system serves as the environment for the BP system. In this case the BP system is modelled by a DES-P model and the ICT environment is modelled by a TFA model. The communication between cooperating DES-P and TFA models is denoted by number 5 in Figure 1. (Of course, the question of change

of representation and interpretation occurs in this case, too.)

System-Model Matrix of Cooperation for Preliminary Modelling

Preliminary modelling is a very important approach in the simulation process of complex systems. Preliminary models and results may inform the users about the quality and scope of final results and may introduce the possibilities of the simulation method in an early stage of the simulation project. It may support efficient decision making on the objectives of the project, to get information about the necessary inputs (it may help to define the data collection need more precisely, to identify and analyse sources of data), to reveal the direct and potential users of simulation results (it may help in the identification of the set of systems influenced by the simulation).

![Figure 2 The System-Model Matrix for Preliminary Modelling](image)

As for preliminary modelling, it is the best to apply the principle of parsimony (Pidd 1991); i.e. that the final simulation model should be built in incremental steps starting from a non-complicated model.

In the preliminary modelling stage, we have only a preliminary view of the criticalness of a system. For preliminary categorisation of systems we use the proto-critical and proto-non-critical classification.

In the system-model matrix for preliminary modelling (Figure 2) we have only two different types of rapid models: we have TFA models to model proto-critical parts of ICT systems and EFA models to model proto-critical parts of BP systems and contracted TFA and EFA models to model proto-non-critical parts of ICT and BP systems respectively. Between cooperating TFA and EFA models there is a change of interpretations in communication (number 2 in Figure 2). In the cooperation between contracted and non-contracted models there is no change of representations in communication (number 1 and 3

---

1 The term contracted will be defined in section “Designing Cooperation of Models: Preliminary Modelling”
in Figure 2). (The reasons will be explained later.) Connections number 4-6 are theoretically possible, but have no practical significance in preliminary modelling.

**DESIGNING COOPERATION OF MODELS: TRANSLATION OF CONCEPTUAL MODELS**

**Conceptual Modelling**

Conceptual models planned to be used as a model design support tool (Muka and Lencse 2006) in different phases of simulation projects (in our examination: mixed simulation projects); in the phase of preliminary modelling (“Defining goals” and “Gathering and analysing data” points of the simulation method) and also in the phase of detailed modelling (“Model design and model building” point of the simulation method). Therefore, it is crucial how the information acquired in conceptual modelling can be translated into the set of cooperating rapid and detailed models.

First, let us summarise the essence of conceptual modelling: The central idea of SSM (Soft Systems Methodology, Checkland 1989) is the conceptual model. The outcome of SSM may also be used for information system analysis and design (Curtis 1989; Gregory 1993).

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.

In our previous paper (Muka and Lencse 2006), we have described the modified conceptual modelling (MCM). MCM may be characterised as an extension of SSM models with extra features and grafting the methods of using extended models into SSM. MCM, which is focusing on the design of information systems in an enterprise, is applicable both on soft-system level and on hard-system level of simulation model design (Muka and Lencse 2007a).

In an MCM, a key activity is performed generally by a Business Process (BP) function or by an ICT system function, that is, any function in an enterprise can be performed by some relevant business process (P subsystem) with its human resources or by some relevant ICT system (IT subsystem) with its technical resources. Thus, MCM elements can be P-type or IT-type depending on what they represent, BP or ICT system function. (This is the basic feature of MCM that makes it applicable in mixed simulation projects.)

An important feature of MCM is that any IT element in the model should be connected to at least one P element in order to have its human resource connection.

For the analysis of necessary and sufficient conditions in MCM, three element types are used: 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 a condition function necessary to perform the basic function while A is an agent element ensuring the sufficiency (“motivational”) condition for the basic function to be completed.

**Virtual time in modified conceptual models 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 (Muka and Lencse 2007b).**

**Interpretation of MCM Analysis Results**

MCM analysis, which is a complete set of methods grafted into SSM, is used for system analysis. After finishing the MCM analysis, the designer has a lot of information about the systems to be modelled and simulated, which may be summarised as follows:

- There has been built a set of conceptual models of IT and P subsystems, the functions of subsystems has been described and the type (A, F, and C) of every subsystem has been determined.
- The connections between subsystems have been identified.
- The virtual time system of subsystems has been assigned and synchronised.
- The execution time of subsystems has been found out (as a result of the time decomposition) together with the fitting of execution time of different types of subsystems and the set of critical IT and P subsystems have been defined.
- For the time decomposition, a simulator may have been used and pre-assumptions on sequential-parallel simulation of subsystems may have been produced.
- The resolution of the conceptual model has been set (completing the resolution increasing-decreasing transformations).

At this point, before starting the translation of conceptual models, the question whether all the IT subsystems are explicit should be decided (P subsystems may contain hidden IT subsystems when extended).

**Problems of Translation: A Customer Request Processing System**

In the following, by describing the example of a Customer Request Processing System (Figure 3), the questions of translation of results of MCM analysis into ICT and BP models will be examined.

In the example, there is a Customer Help Desk, which handles the requests of users (using different ICT systems) and decides about the involvement of the Service Department. The Service Department processes and schedules the requests from the Customer Help Desk and direct alarm requests from users, also using ICT systems. The VoIP Service Department provides VoIP services to these systems according to the requirements of the service level agreement (SLA).

The conceptual models in the example are only conceptual model fragments, and do not try to present all the details of the systems. There can be many questions that can be examined in the introduced situation. For example:
- Optimisation of resources used in Service Department, taking into account the requirement of increased user satisfaction
- Efficient division of work between an “Automatic customer request answering and processing software system” and the operators of the Service Department
- The required change in the structure of the Service Department to achieve the shortest reaction time
- To determine an appropriate SLA (Service Level Agreement) for VoIP between Service Department and the departments using the services
- Satisfactory and efficient intranet capacity for green number of services, VoIP services and direct user alarm services

Now, let us see the example MCMs in detail:

In Figure 3, the Customer Help Desk receives a Customer Request from a General User. The “Automatic request answering and forwarding” (ITF₁ subsystem) of the Customer Help Desk handles the request using IT subsystems (ITC₁₁ and ITC₁₂) providing the necessary service conditions. In the next fragment of Customer Help Desk model the customer request that was forwarded by the “Automatic customer request answering and forwarding” subsystem is received by help desk operator (PF₁ subsystem) using VoIP (condition provided by ITCᵥ subsystem) and customer information asked from CRM (Customer Relationship Management) (PCᵥ₁ subsystem) to continue processing by subsystem PFᵥ₁ and to decide about involving Service Department. PFᵥ₂ may involve Service Department using VoIP services (Transient 1).

The Service Department receives Transient 1 (subsystem PCᵥ(i+1)₁₂) and processes the request from Customer Help Desk (subsystem PFᵥ₁). The PFᵥ₁ subsystem of the Service Department manages the resource pool (using a Workflow Management System provided by PCᵥ subsystem).

The Direct User is able to send direct alarm request to the Service Department. The request (user alarm signal in this case) is received by PCᵥ(i+1)₁₃ subsystem.

Let us notice the difference: a General User reaches the Help Desk through a green number service (subsystem ITCᵥ in the MCM of the General User) but a Direct User has a special subsystem (subsystem ITCᵥ(i+1) in the MCM of the Direct User) for the access.

The VoIP Service Department is responsible for granting the VoIP service for the Customer Help Desk and for the Service Department. In the MCM of the VoIP Service Department shown model fragments are shown of P (PFᵥ and PFᵥᵢ) and IT (ITAVᵢ, ITCᵥ, ITCᵥᵢ) subsystems of the system providing VoIP services.

![Figure 3 Fragments of the Modified Conceptual Model of a Customer Request Processing System](image-url)

**P subsystems with Hidden IT systems**

- The PCᵥᵢ₁ conditional subsystem provides the necessary customer information for subsystem PFᵥᵢ for request processing. In the expanded PCᵥᵢ₁ subsystem there are two subsystems: the PCᵥᵢ₁ process subsystem and the ITCᵥᵢ₁ IT subsystem. Both subsystems have their own output that means that PFᵥᵢ₁ may send direct request to the CRM database or in case of a more complicated inquiry the CRM operators may be asked.
- The situation is similar in the case of PCᵥ in the Service Department, where PFᵥ uses a Workflow Management System for resource pool management.
- The PCᵥ(i+1)₁ subsystem receives the alarm signal from a Direct User. Of course, in this subsystem there should be an ICT part (the Direct User has
direct access to the Service department, through for example some devices connected to a leased line) but there is a special process part, too.

- The $P_{12} - P_{11}$ connection uses VoIP services (see expanded $P_{11}$ subsystem in MCM of Service Department). The number of channels (rate of successful calls), which may have influence on the reaction time of the whole Customer Request Processing System, may be important for this connection.

**Determining ICT-BP Interactions in Conceptual Models**

By examining conceptual models, the *interactions* between planned simulation models of ICT and BP systems can be identified.

1. An activity of the BP system uses the ICT system:
   - $P_F$, may be performed if $P_C$ is provided
   - $P_{F_{11}}$ may be performed only if $P_F$ asks for information from CRM and the necessary information is obtained by CRM ($P_C$)
   - $P_F$ may be performed with the condition provided by $P_C$

2. An activity of the BP system sends information to another activity of the BP system using the ICT system:
   - The $P$ subsystem of Customer Help Desk ($P_{12}$) sends a message to another $P$ subsystem of service Department ($P_{11}$) using the IT subsystem $IT_{C_{12}}$

3. The ICT system acts as an initiator towards the BP system:
   - A user alarm generated by subsystem $ITF_{a1}$ of the Direct User generates entities for the processes modelling $P$ subsystem $P_{C_{11}}$
   - $ITF_1$ acts as initiator for $P_F$ (and may also for other subsystems)
   - The ICT system acts as an initiator towards the BP system:

4. The BP system acts as an initiator towards the ICT system:
   - The $P$ subsystem $P_{a}$ of General User acts as an initiator for $ITF_1$ of Customer Help Desk

**Remark:**

Of course, depending on the designers view, the type of interaction may change. For example: $P_{a}$ of a General User may send a message to $P_F$ of Customer Help Desk using IT subsystems ($IT_{C_{12}}, ITF_1, IT_{C_{11}}, IT_{C_{12}}, IT_{C_{10}}$ and others).

**Points to Consider for Translation**

The following points that can help in translating decisions:

Critical F subsystems (or a C subsystem evaluated as critical in the MCM analysis) will probably be translated into detailed ICT or BP models.

Those elements that are part of critical connections can probably be translated into detailed ICT or BP models.

Non-critical (including environmental C (and A)) subsystems will be translated into TFA and EFA models or their function may be approximated statistically if it satisfies the accuracy requirement.

P elements in the set of critical subsystems or connections should not contain hidden IT elements, that is, these P subsystems should be expanded during the translation.

**DESIGNING COOPERATION OF MODELS: RAPID AND DETAILED MODELS**

DES-IT and DES-P models are detailed models of ICT and BP systems, thus they are typically applied in detailed modelling phase of simulation projects.

For the cooperation of DES-IT and DES-P models (connection number 2 in Figure 1), it is not enough simply to switch from messages in DES-IT to entities in DES-P, but it is necessary to change interpretations between them. To develop the change of interpretations the basic interactions are identified (number 1-4 in “Determining ICT-BP Interactions in Conceptual Models”) between DES-IT and DES-P models.

If it is useful to divide the ICT system into critical and non-critical parts the TFA method is used to model the non-critical part. During the cooperation of DES-IT and TFA model (number 1 in Figure 1) there is a need for the bidirectional change in representations that is a conversion between different traffic representations — messages and statistics. In the DES-IT segment it is necessary to collect the appropriate statistical characteristics of the message flow (which can be characterised by e.g. message length, inter-arrival time, the sources and destination of the packets, etc.) in order to produce the statistics that are used in TFA. In the direction from the TFA model to the DES-IT model messages should be generated on the basis of statistics (traffic model in TFA).

The same consideration can be taken into account concerning DES-P and EFA models (number 3 in Figure 1), where only an entity flow is used in DES-P instead of message flow in DES-IT and entity-load model in EFA instead of traffic model in TFA.

Of course, if there are TFA and EFA models in the model of the whole system then there can be direct cooperation between them (number 4 in Figure 1). The rules of cooperation between TFA and EFA models may be found using communications for traffic representations (1 and 3 in Figure 1) and connection for change of interpretations (2 in Figure 1). These rules can be used during simulation (Lenese and Muka 2007a).

If there is only one non-critical part in the model (TFA or EFA) (asymmetrical non-critical situation) then communication number 6 or 5 should be used in case of existing traffic to DES-P or DES-IT. The rules of cooperation may be revealed using the approach from the previous symmetrical case, fitted to this situation.

If there is a model (DES-IT or DES-P) of a critical system and a model of its non-critical environment (EFA and TFA respectively) in the model then only communication used for cooperation is 6 (or 5 respectively) in Figure 1. This cooperation may be prepared in the way described previously, in the case of asymmetrical non-critical systems.

When is it worth modelling ICT and BP systems together?
Let us see some examples:

- **BP model in ICT model**: The BP system (with a non-predictable behaviour) is an important traffic source for the ICT system (for example customer service offices in the ICT infrastructure).
- **ICT model in BP model**: for example the optimisation of proportion of automatic (produced by an answering software) and operator performed activities of a help desk system.
- **ICT and BP model**: for example to set up the most efficient SLA (Service Level Agreement) for a service department of a complex ICT system.

If there is a close relationship between the ICT and BP models the best way is to organise them into one process or if not it is possible then to find an efficient way of synchronisation of the models.

If there is a TFA (or EFA) model in the model (to model non-critical parts of the system) it can be synchronised with the critical DES-IT (or DES-P) model in the ways described in (Lencse 2004).

In case of asymmetrical non-critical situation or if there is a critical DES-IT (DES-P) model and a non-critical EFA (TFA for DES-P) in the model of the system, the above mentioned methods may be used, but the problem of change of representations and interpretations should not be forgotten.

If there are only TFA and EFA models in the model of the system, the synchronisation seems to be simple (both TFA and EFA are snapshots of operation of the system) but we have only limited possibilities because we do not have information about the models’ virtual time.

The essence of building cooperating models can be summarised as follows. If we have a set of different models and we would like to cooperate any two of them we must build the models in the way that they can cooperate (to produce and send all the information necessary to the other model) or to formulate this principle in a more general way, if there is a chain of cooperation (for example a chain of DES-IT, DES-P, EFA models) the system of conversions in the model chain and the models themselves should be built in a way that they should keep, regenerate and generate all the information necessary in the chain of conversion and models.

**DESIGNING COOPERATION OF MODELS: PRELIMINARY MODELLING**

In the application described in (Lencse 2004), cooperating DES and TFA models are used for different parts of the same a system (DES is used for modelling the critical part TFA is used for modelling the non-critical part of the modelled systems).

In preliminary modelling, which is our case, it is appropriate to use rapid models for modelling the whole system (both for ICT and BP systems and for critical and non-critical parts).

The **testing step** of the TFA/EFA spatial phase (Lencse and Muka 2007b) uses DES models to reveal routing properties of models of subsystems. That is, to perform this step, DES models of subsystems should be built. Keeping to the rule of parsimony we should reuse this DES model: the same **testing phase** is used for collecting statistics about the communication between DES models of ICT (and BP) systems, about message flows in the ICT system and about entity flows in the BP system. The same testing phase collects information for setting up the conversion (interpretation) rules between ICT and BP parts.

The use of MCM time decomposition results about sequential-parallel processing support (Muka and Lencse 2007b) may help to speed up the model building and testing phases.

In (Lencse and Muka 2008) the terms inflated / deflated models were introduced.

In the test phase we use inflated DES-IT and DES-P models, in the model running phase we use TFA and EFA models, which are deflated DES-IT and DES-P models. The deflated models use “inflated communication” i.e. message flow (or entity flow) in virtual time. (The concept of communication is shown in Figure 4.)

![Figure 4 The Concept of Connecting TFA (or EFA) Models](image)

A system of deflated models with deflated internal communication we call **contracted model**.

The contracted TFA (or EFA) (Figure 5) models behave as one deflated TFA (or EFA) model.

For the communication of ICT and BP models, the change of interpretations (of message and entity flow) based on interactions between the modelled systems is necessary. Now, let us examine the system of cooperation in preliminary modelling (Figure 5).

TFA<sub>1, n</sub>, EFA<sub>1, m</sub> models are the models of the proto-critical parts of ICT and BP systems respectively. TFA<sub>a, b, c, m, n</sub> (or EFA<sub>a, b, c, m, n</sub>) is the model of a key activity produced by a proto-critical ICT (or BP) subsystem “a” (or “b”) (of course TFA<sub>a</sub> (or EFA<sub>a</sub>) may be a contracted, proto-non-critical model too).

---

2 RDT (Routing Decision Tree) building phase (Muka and Lencse 2007b)

3 In a nutshell, if a compound model is inflated, its internal structure as well as the operation of its parts are modelled in detail. If it is deflated, we omit its internal structure and operation and imitate its behaviour for the outside world by a simpler algorithm that acts similarly but of course not completely the same as the original compound module.
The non-contracted TFA/EFA models may be built as follows:

1. Set up the required model resolution using MCM resolution transformations (grouping, integrating).
2. Create DES models of subsystems.
3. Collect statistics in the TFA/EFA test phase between subsystems of the proto-critical systems and between the proto-critical and proto-non-critical system parts, too.
4. Set up \( S_{RU} \) (size of routing unit, (Lencse and Muka 2007b)) for each TFA/EFA subsystem model.

The contracted TFA/EFA models may be built as follows:

(Contracted TFA/EFA models are the models of the environment. One contracted TFA model is to be built for the ICT environment and one EFA model for the BP environment.)

1-3. The steps from 1 to 3 are the same as the steps of building non-contracted models.
4.Contract the model: Make one TFA/EFA model from subsystem models. While each element of the system of the non-contracted TFA/EFA models (TFA\(_{m} \), EFA\(_{m} \)) has its own virtual time, the contracted part uses the same virtual time (and the same \( S_{RU} \)).

The operation of the contracted and non-contracted TFA/EFA models is coordinated by DES\(_{COORD} \)'s functions. A DES\(_{COORD} \) performs the following tasks:

- Manages the virtual time system of models (starting from the conversion of the MCM virtual time system).
- Manages the connection map and the communication of TFA/EFA models.
- Controls the work of the contracted and non-contracted models. (Contracted and non-contracted TFA/EFA models of the preliminary model work in the same way.)

- Manages the communication (bidirectional interpretation) between ICT and BP parts.

The preliminary model with cooperating TFA and EFA models works as follows (while the TFA/EFA runs the virtual time will not elapse, but the result will have influence on the virtual time of the whole process):

- At \( t_{0} \) DES\(_{COORD(TFA)} \) and DES\(_{COORD(EFA)} \) prepare starting conditions and starting messages:
  - After message → statistics conversion a set of models from TFA\(_{m} \), and contracted TFA may be started to run by DES\(_{COORD(TFA)} \) and a set of models from EFA\(_{m} \), and contracted EFA may be started to run by DES\(_{COORD(EFA)} \).
  - An interpretation cycle may be started (messages → interpretation → interaction → messages → statistics).

- At \( t_{0} + \Delta t_{1} \) the selected TFA (or EFA) model starts running.
- At \( t_{0} + \Delta t_{1} + \Delta t_{2} \) the selected TFA (or EFA) model returns statistics.
- At \( t_{1} + \Delta t_{1} + \Delta t_{2} \) DES\(_{COORD(TFA)} \) and DES\(_{COORD(EFA)} \) starts statistics → message conversion.
- At \( t_{1} \) information interchange may be initiated between ICT and BP models (if necessary).
- At \( t_{1} \) DES\(_{COORD(TFA)} \) (or DES\(_{COORD(EFA)} \)) gets back interaction information and starts interaction → interpretation → messages → statistics conversion.
- And so on.
- Finish when the time for examination elapsed or the answer is produced.

ICT and BP may run in two processes:

- ICT part in one process, BP part in another process.
- A suitable synchronisation algorithm is used between ICT and BP parts.

CONCLUSIONS

This paper addresses the problem how the system of cooperating rapid and detailed modelling and simulation methods can be used to increase the efficiency of mixed simulation projects initiated to support the design of interconnected ICT (Information and Communication Technology) and BP (Business Process) systems.

First, the general features of cooperation of rapid and detailed modelling methods in the phase of preliminary and detailed modelling are introduced using system-model matrices. Both ICT and BP systems may have critical and non-critical parts in the detailed modelling phase and respectively proto-critical and proto-non-critical parts in the phase of preliminary modelling.

The next part describes the method of translating information obtained in the phase of conceptual modelling for building the system of cooperating rapid and detailed models. After a brief summary of general aspects of conceptual modelling, the essence of using MCM (Modified Conceptual Modelling) in mixed (ICT and BP) simulation projects (including general aspects of conceptual modelling) is explained. An example is shown to introduce the tasks of translating the results of MCM analysis by analysing the
problem of hidden IT systems and the problem of determination of ICT-BP interactions.

Then the questions of cooperation in the phase of detailed modelling are analysed. An evaluation is given of the methods of representation, interpretation and synchronisation of cooperating models. Two new methods are introduced: how to use rapid BP models that function as environment in detailed modelling of ICT systems and also how to use rapid ICT models functioning as environment in detailed modelling of BP systems.

Finally, a novel method for preliminary modelling based only on cooperating system of rapid models is described. In the proposed method a contracted rapid model is used to model the proto-non-critical parts of the system and a set of rapid models is used to model the proto-critical part of the system. A coordination element is used to deal with timing and conversion problems (messages and statistics generation and control of interaction of systems).

We conclude that the proposed modelling methods based on cooperating rapid and detailed models highly increase the efficiency of the performance evaluation of interconnected ICT and BP systems in the preliminary and in the detailed modelling phase.

REFERENCES


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 info-communication systems. Since 1997, he has been working 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 Infrastructureal 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 has been working 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 Infrastructureal Systems at the Széchenyi István University of Győr.
MODELLING OF GAS TURBINE BASED PLANTS DURING POWER CHANGES

Henryk Kaproń*, Michal Wydra**
Faculty of Electrical Engineering and Computer Science Lublin University of Technology
Nadbystrzycka 38A, 20-618 Lublin, Poland.
*E-mail: h.kapron@pollub.pl
**E-mail: m.wydra@pollub.pl

ABSTRACT
A combined cycle power plant, which combines a gas turbine and steam turbine, can achieve high energy efficiency. Now many combined cycle power plants have been installed in the world, because of it’s fast response and primary fuel saving. Fast response to power change of CHP plant is important when system frequency drops or when system is near blackout situation, however this paper deals with minimization of fuel consumption of CHP plant under normal operating conditions according to scheduled power changes and predicted ambient temperature and pressure. Main operational cost of CHP plant is fuel price and it’s usage, which can vary about 70% of all costs. Presented analysis uses modified dynamic model of single-shaft combined cycle plant for simulations of fuel usage in particular ambient conditions and different power change gradients.

INTRODUCTION
A combined cycle power plant has high energy efficiency which can exceed 50% of electrical efficiency, and if a particular plant, have installed a heat exchanger for producing heat, energy efficiency rises to about 75%. Nowadays every unit wants to use the minimum amount of fuel, because it’s cost can be extremely high, about 70-80% of total production expenses.

In this paper, an analysis is based on modified dynamic model of single-shaft combined cycle plant which was developed before for studying dynamic response of CHP plant during frequency drops. Change of output power of a plant usually is made slow according to gas and steam turbine time constants, so the output power change can be assumed as quasi-static, and for more accurate simulations the full dynamic model is used. Because total efficiency of a plant strongly depends on ambient temperature and pressure, a one day ahead numerical weather report was used for simulations. Authors believe that dynamic model can be more accurate. It was developed by combining three models well described and simulated in past few years, connected with grid frequency deviation and response of CHP plants. Plant modeling using differential equations or equivalent transfer functions, can be efficient and if then classic optimization is used minimal, energy power change trajectory for CHP plant can be derived.

MODEL OF COMBINED CYCLE PLANT
Single-shaft power plant consists of a compressor, combustor, gas turbine, steam turbine, waste heat recovery boiler, steam turbine and generator is shown in the Fig.1. The plant Fig.1 and model Fig.2 is shown below, it was used and modified as follows: rotating shaft speed is constant with accordance to normal operating conditions and absence of system frequency deviations.

Figure 1 Single-shaft combined cycle plant

The total electrical efficiency changes as ambient temperature and pressure changes, and also depends on power reference. The gas turbine products about two thirds of total power output, and the steam turbine produces the rest. Energy production in steam turbine is simplified and derived form exhaust temperature $T_e$ and gas flow $W_s$. The gas turbine unit, and its production of energy was modeled based on gas temperature changes in compressor, combustor and expander. The following equations were used to obtain power output of gas and steam turbine. Since the air is adiabatically compressed, compressor discharge temperature $T_d(K)$ is given as follows:

$$T_d = T_e \left(1 + \frac{x-1}{\eta_e} \right)$$

(1)

$$x = \left(\frac{P_m}{W_s} \right)^{\frac{\gamma-1}{\gamma}}$$

(2)

where $T_e(K)$ is compressor inlet temperature and equals ambient temperature, $\eta_e$ is compressor efficiency, $x$ is compressor temperature ratio, $P_m$ is design compressor pressure ratio, $W_s$ is airflow in per unit of its rated value, $\gamma$ and is ratio of specific heats. The airflow $W$ depends on the ambient temperature $T_d(K)$ and the atmospheric pressure $P_a$.

$$W = W_s \frac{P_d}{P_{ao}} \frac{T_{ao}}{T_d}$$

(3)

where subscript “o” denotes rated value, and $W_s$ is the airflow at $P_{ao}$ and $T_{ao}$. Gas turbine inlet temperature $T_i(K)$ is given by

$$T_i = T_d + \left(T_{io} - T_{ao} \right) \frac{W_s}{W}$$

(4)
where $W_f$ is fuel flow in per unit of its rated value. The fuel flow is a negligible amount compared with the airflow. Gas turbine exhaust temperature $T_e(K)$ is described as follows:

$$T_e = T_f - \frac{1}{\eta_f} \left[ 1 - \frac{1}{x} \right] \eta_e$$

(5)

where $\eta_f$ is turbine efficiency. The exhaust gas flow is practically equal to the airflow. With these values, the net energy supplied to the gas turbine is given by

$$E_g = K_0 \left( T_f - T_e \right) - (T_d - T_i) W$$

(6)

where $\{K_0\}$ is a constant. This energy is converted to the power $P_g$. The energy collected by the heat recovery boiler and the steam turbine is where $\{K_1\}$ is a constant, identified before.

$$E_s = K_1 T_s W$$

(7)

The model have IGV- exhaust temperature controller which changes the reference temperature after gas turbine in the function of ambient temperature, which was main modification of the model. The temperature control reference value $T_{ra}$ depends primarily on the compressor discharge pressure, which has been assumed to be constant in this model. The control point then becomes a linear function of ambient temperature $T_a$:

$$T_{ra} = T_f - 0.6(T_{ISO} - T_a)$$

(8)

where: $T_f = 5320C$ is rated exhaust temperature form turbine, $T_{ISO} = 300C$. The static characteristic of mentioned model is shown in the Fig. 3.

**CASE STUDY**

The single-shaft combined cycle plant is considered in shown material, and typical parameters of the unit are: 11000C class of gas turbine cycle, its rated power output is 160 MW where 106.7 MW is produced by gas turbine and 53.3 MW by steam turbine. The problem to solve is how an operator should change unit’s power to minimize fuel consumption in particular ambient conditions. These conditions are usually predicted by numerical weather forecast, and now they are sufficient precise. The analysis assumes CHP load change with gradients from 1 MW/min to 7.5 W/min, which are technical bounds of unit. Input signals to combined-cycle plant model are: power reference trajectory $P_{ref}$, ambient temperature $T_a$, ambient pressure $P_a$. The Operational power change is made with accordance to unit’s power schedule. Fig. 4 shows the concept of simulations. The outputs of plant’s model are the total power output of unit and integral from fuel flow in the time domain which is actual fuel consumption. All simulations were made in Matlab/Simulink.

**Figure 4 Power change concept with different gradients according to unit’s load schedule**

The Fig. 5 and Fig.6 shows the example changes of ambient temperature and pressure used for simulations of change load with different gradients, and results of fuel usage when an operator wants to rise output power from 50% to 70% of unit’s power is shown in Fig. 7. The differences in fuel usage reduce operational costs of the plant. It can be also observed that simulations considered only one change, and if there is a more power changes, cost reduction can grow.

**Figure 5 Example ambient temperature change (6 hours)**

where rated total electrical efficiency is 49%.
CONCLUSIONS AND SUGGESTIONS

Analysis of simulation results shown that the higher gradient, when rising output power the less fuel is used. When lowering power, the more radical descent the less fuel used. This simple analysis is theoretical and need to be verified on real CHP plant, however conclusions seems to be correct. The faster plant get better efficiency, the less fuel it will use. Concluding second scenario if the plant will remain longer at higher efficiency less fuel is used. Described method of minimizing fuel consumption can be spread for all day power change schedule, and all day ambient temperature and pressure numerical forecast is necessary, but then every change of output power should have it`s own optimal change gradient. This last condition also can be fulfilled because parametric optimization now is common. The main problem is to develop highly accurate plant model, and forecast ambient conditions as well. Power change trajectory was considered as ramp with constant gradient, however mentioned trajectory can be derived for every ambient conditions change e.g. every minute.

REFERENCES


BIBLIOGRAPHIE

HENRYK KAPROŃ, Professor at the Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Chief of Power Generation an Economy Department. From 1995, prof. Kaproń has been come to the CHP Lublin-Wrotków Ltd. as a member of control exploitation 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. MICHAL WYDRA , Assistant Lecturer at the Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Automation and Measurement Dept. Currently working toward the Ph. D. degree. Interested in advanced modeling and control techniques applied to power plants.
SIMULATION IN BIOLOGY
THERMOREGULATION IN P. TERRAENOVAE AGGREGATIONS, AN AGENT-BASED APPROACH

Gildas Morvan, Daniel Jolly
LGI2A EA 3926
Université d’Artois
Faculté des Sciences Appliquées
Technoparc Futura F-62400 Béthune
email: {gildas.morvan, daniel.jolly}@fsa.univ-artois.fr

Damien Charabidze
Laboratoire d’Entomologie
Institut de Médecine Légale
Université de Lille 2
Place de Verdun F-59035 Lille
email: damien@forenseek.org

KEYWORDS
agent-based simulation, thermoregulation, forensic entomology

ABSTRACT
This paper deals with an important issue of forensic entomology: the estimation of the temperature in a "maggot mass". An agent-based model of the thermoregulation behaviour of fly (Protophormia terraenovae) larvae is described. Simulation results show that the model reproduces an important observed pattern and thus, might be used in entomological expertise.

INTRODUCTION

Forensic entomology is widely used in criminal investigations to determine post-mortem interval (PMI) and possibly information about the crime such as death location (Marchenko 2001). This is done studying the entomofauna, i.e., the insects, mainly maggots (fly larvae), found on the corpse. A PMI is usually estimated by experts using analytic models of insect development. These models can be easily employed to perform retrodictive reasoning but do not take into account the ecosystemic context. Thus, datings performed using these methods are sometimes overestimated and not as precise as they could be. An important cause of inaccuracy is the thermoregulation behaviour of larvae. This phenomenon, known as the "maggot mass effect" has been described in numerous publications. However, no model has been proposed to handle this issue. In this article, an agent-based model of the thermal behaviour of fly larvae is introduced. A species of forensic interest, Protophormia terraenovae, is considered. First, forensic entomology methods and issues are introduced, then, the thermoregulation model is described, as well as its implementation. Finally, simulation results are detailed.

INTRODUCTION TO FORENSIC ENTOMOLOGY

Overview of the colonization process of a corpse
A cadaver is an important source of nutrients for insects, especially for necrophagous species. Many species will succeed on the body, each one of them being interested by a particular stage of decomposition. The first necrophagous species to colonize a cadaver are Diptera but in advanced stages it is common to find Coleoptera species. A few hours after the death, the first Diptera females are attracted by the body and lay eggs on it, mainly on natural orifices. Once the eggs hatch, Diptera larvae colonize the body to feed. Three larval stages, or instars, named L1, L2 and L3 will succeed. When a larva ends its development cycle, it leaves the corpse to find a suitable place for pupariation, an inactive stage at the end of which the larva eventually turns into an adult fly (figure 1).

Figure 1: Development cycle of a Diptera.

As many living organisms, the development of Diptera is temperature-dependent (Stinner et al. 1974). Thus, the development speed \( \frac{da}{dt} \) of an individual is given as a function \( f \) of the temperature \( T \) varying in the time \( t \)

\[
\frac{da}{dt} = f(T(t)).
\]

On this basis, various models have been developed, some
assuming a linear relation between development speed and temperature, some a more complex relation. A review of development models can be found in Wagner et al. (1984). The validity of those models will not be discussed in this paper.

Methodology of entomological expertises

When a cadaver is discovered, investigators take samples of eggs, larvae or pupae from the body. Entomologists determine the species and the accumulated rates of development (denoted Δa) of the oldest individuals. Then, for each one of them, the laying time t1 (generally close to the time of death) can be calculated from the equation

\[ Δa = \int_{t1}^{t2} f(T(t))dt, \]

where t2 represents the time of the cadaver discovery. 

Data from the nearest meteorological station are used in order to estimate the temperature. However, considering that the temperature at any point of the body is equal to the temperature recorded by the nearest meteorological station is not exact for three reasons. Firstly, ecosystem specificities can radically influence the temperature around the body as shown in Gosselin et al. (2006). Secondly, the thermal inertia of a corpse is important, especially in the first hours after death (Henssge 2004). Thirdly, the heat generated by larval aggregations can raise the temperature locally up to 20°C. This phenomenon, known in the literature as the "maggot mass effect" and its consequences on PMI estimation have been discussed by many authors (Marchenko 2001, Slone and Gruner 2007). Thus in many cases, entomological expertise results are inaccurate and given with an important margin of error.

The agent-based model presented in the next section focuses on the third cause of inaccuracy. Indeed, this model allows to simulate the thermoregulation behaviour of large masses of Diptera larvae.

Behaviour of Diptera larvae

The behaviour of Diptera larvae remains mostly unknown. The present study is based on observations performed by the authors and found in the literature, e.g., in Hobson (1931). It is well known that Diptera larvae have a gregarious behaviour that favours the formation of huge aggregations. This behaviour is not species or instar specific. Moreover, there is no discrimination between instars or species in aggregations. It has been shown that larvae are attracted by a stimulus emitted by conspecifics. Though the nature of the stimulus (vibrations, ammonium hydroxide, etc.) has not been clearly identified, we assume that this stimulus can be represented by a pheromone-like variable. The dynamics of larvae inside an aggregation is very complex. However, the observation of larva masses revealed a precise repartition of individuals, depending on their physiologic state. Thus, it is possible to distinguish between three main behaviours:

- **feeding**: larvae are immobile, vertically aggregated and in direct contact with the nutritive substrate;
- **looking for food**: larvae which the crop is empty try to reach the centre of the aggregation, i.e., to access to the food;
- **digestion**: larvae which the crop is full move around the aggregation, looking for optimal temperature.

The structure of a maggot mass can be explained from these three simple behaviours. While there is no collective decision or space optimisation, a coherent and self-organised behaviour emerges from local interactions between larvae. This gregarious behaviour can be regarded as a primitive, though very efficient, form of collaboration. Indeed, it allows larvae to share digestive fluids and optimise the temperature in the aggregation. Each larva emitting heat, due to its metabolism and rubbings, it can result an important, quick and local increase of the temperature. Thus, larvae in an aggregation speed up their development period and maximise their chance of surviving.

**DESCRIPTION OF THE MODEL**

**Model structure**

The model has been implemented on TurtleKit (Michel 2002, Michel et al. 2005), a logo-like simulation platform built on the generic multi-agent platform MadKit\(^1\) (Ferber and Gutknecht 1998). In TurtleKit, simulation agents act in a two-dimensional environment discretized into "patches". A patch is then an homogeneous environment portion to which "patch variables" are attached. Simulation agents can perceive and act in the environment by manipulating patch variables. However, patch variables can have their own dynamics. The behaviour of a simulation agent is defined as a finite state automaton, each state representing an atomic behaviour. Moreover, the TurtleKit platform provides graphical monitoring tools.

Our model environment has been designed to imitate the environment used in the experiments performed at the entomology laboratory of the medico-legal institute of Lille. Three variables are attached to patches: the amount of food available, the concentration of a pheromone-like substance emitted by larvae and the temperature. The thermal dynamics of the environment has been modelled using a cellular automaton approach presented in Veremne et al. (2008). Simulation agents represent P. terraenoviae individuals during

\(^1\)http://www.madkit.org
their preimagal life, \textit{i.e.}, until they become adult flies. The development model implemented in the simulation agents is described in Stinner et al. (1974). A model of development variability inside the larva population based on Régnière (1984) is also used. The crawling speed of larvae has been estimated using the model presented in Charabidze et al. (2008). In the next section, the architecture of simulation agents is presented. These agents are purely reactive, \textit{i.e.}, they do not have any symbolic representation of their environment or the other agents. Their behaviour depends on environmental stimuli and their internal state. Thus, they can be characterised as “drive-based agents” (Ferber 1999).

**Architecture of the agents**

The main idea behind this model is that the behaviour of a larva is different whereas it is in viable or non-viable conditions and that its behaviour in non-viable conditions regulates the temperature of the aggregation by optimising its density. The architecture of the simulation agents can then be represented as a 2-state automaton (figure 2). The state \( b_1 \) represents the behaviour of the agent in viable conditions whereas the state \( b_2 \) represents the behaviour of the agent in non-standard, \textit{i.e.}, non-viable or "stress" conditions. The transition between these two states depends on a condition \( c_p \). Here, \( c_p \) is defined as follows:

\[
c_p : T > T_{max},
\]

where \( T \) is the temperature felt by the agent and \( T_{max} \) is the maximal temperature supported by a larva.

\[\neg c_p \rightarrow b_1 \rightarrow c_p \rightarrow b_2 \rightarrow c_p\]

Figure 2: Behavioural model overview. \( b_1 \): ”normal” behaviour, \( b_2 \): ”panic” behaviour.

In the state \( b_2 \), the agent moves randomly in the environment. The ”normal” behaviour is a little more complicated and can be defined as a 3-state automaton representing the three standard atomic behaviours of a larva (figure 3).

A variable \( j \) proper to each agent, is introduced to represent the state of the crop of the larva (\( j = 0 \) if the crop is empty, \( j = 1 \) if the crop is full). Moreover, we assume that the feeding speed is a function of the temperature \( T \), \textit{i.e.},

\[
\frac{dj}{dt} = \begin{cases} 
0 & \text{if } T \leq T_{min} \\
1 & \text{if } T \geq 25 + T_{min} \\
\frac{T-T_{min}}{25} & \text{otherwise,}
\end{cases}
\]

where \( T_{min} \) is the minimal temperature at which a larva is active.

Each state of the automaton represents a specific behaviour. The state \( b_{1,1} \) represents the feeding behaviour: the agent does not move and consumes environment resources. An agent feeds on a given patch if its crop is not full (\textit{i.e.}, \( j < 1 \)) and a condition \( c_e \) is true. \( c_e \) can be intuitively defined as follows: the patch where the agent is located is a local maximum for the maggot signal and the nutritional capacity of the patch is superior to a threshold, \( n_{min} \).

The state \( b_{1,2} \) represents a temperature optimisation behaviour. After feeding, the larva looks for the patch that optimises its development speed \( f(T(t)) \). The algorithm that underlies that behaviour is a simple gradient descent. The agent keeps on that state until its crop is below a given value, \textit{i.e.}, \( j < \gamma \).

\[
\begin{align*}
(j < 1) \land c_e & \quad b_{1,1} \\
(j = 0) \lor c_e & \quad b_{1,3} \\
(j < 1) \land \neg c_e & \quad b_{1,2} \\
(j > 0) \land \neg c_e & \quad j < \gamma \\
& \quad j \geq \gamma
\end{align*}
\]

Figure 3: Behavioural model of a larva in standard conditions. \( b_{1,1} \): feeding behaviour, \( b_{1,2} \): temperature optimisation behaviour, \( b_{1,3} \): food searching behaviour.

Then, the agent switches to the state \( b_{1,3} \) which represents a food searching behaviour. In this state, we assume that larvae try to aggregate with conspecifics. Similarly to the state \( b_{1,2} \), the algorithm that underlies that behaviour is a gradient-descent.

The perception accuracy of the agents is not optimal. Thus, perception levels are determined for the patch variables used in the gradient-descent algorithm. \textit{E.g.}, for a given patch variable, let \( v \) be the patch variable value, and \( l \) its perception level; it means that the agent is not able to differentiate between \( v \) and \( v \pm l \). Let \( l_t \) and \( l_a \) be the perception levels for the temperature and aggregation stimuli.

The temperature elevation \( T_e \), resulting from the activity of a larva on a patch during a simulation step \( s \) (one minute in the simulations presented in the next section), is defined as follows:

\[
T_e(s) = \lambda \cdot \Delta a(s),
\]
where $\Delta a(s)$ is the accumulated rate of development of the larva at the step $s$, and $\lambda$ a parameter of the model. The parameters of the model in "normal" conditions (state $b_1$) have been identified through a qualitative calibration process. The table 1 gives the values determined for these parameters. Perception levels had already been identified in Hafez (1948). However, slightly different values have been determined here.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
<th>unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>0.95</td>
<td>–</td>
</tr>
<tr>
<td>$T_{\text{min}}$</td>
<td>10</td>
<td>°C</td>
</tr>
<tr>
<td>$l_t$</td>
<td>0.5</td>
<td>°C</td>
</tr>
<tr>
<td>$l_a$</td>
<td>$10^{-7}$</td>
<td>–</td>
</tr>
<tr>
<td>$n_{\text{min}}$</td>
<td>$9 \cdot 10^{-3}$</td>
<td>kg</td>
</tr>
</tbody>
</table>

Table 1: Parameters identified in normal conditions

Thus, only two parameters remain to be identified: the maximal temperature supported by a larva, $T_{\text{max}}$, and $\lambda$.

**RESULTS**

Model exploration and calibration

A brute-force exploration of the model predictions has been performed to determine the best values for $\lambda$ and $T_{\text{max}}$. The root mean square deviation (RMSD) is used to measure the difference between predicted and measured values.

![Figure 4: Exploration of model predictions (axis x: value of $T_{\text{max}}$, axis y: value of $\lambda$, axis z: RMSD).](image)

Measured values have been obtained from experiments conducted at the Medicolegal Institute of Lille. *P. terraeovae* larvae were placed without food at the experimental temperature during five hours in order to synchronise their metabolism. Larvae were weighted and placed in a box with sawdust and a piece of beef heart of the same weight than the larva batch. The box was then placed in a thermostatic area programmed at constant temperature. Two temperature probes (precision: ±0.5°C) recorded the temperature inside and outside the box every minute. The maximal temperature increase in function of the number of individuals in the aggregation and the temperature of the environment has been identified as stable pattern that simulations should reproduce. Simulations have been repeated ten times to guarantee the robustness of the results. The mean is used to merge the results of the simulations. The values of the patch variable properties used to simulate the nutritive substrate can be found in Veremme et al. (2008). The figure 4 gives the shape of the two parameter space that has been explored. The table 2 gives the best values found for $\lambda$ and $T_{\text{max}}$.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
<th>unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{\text{max}}$</td>
<td>35</td>
<td>°C</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>$9 \cdot 10^{-4}$</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 2: Values of the parameters of the thermoregulation model identified by simulation

**Simulation results and discussion**

The figure 5 compares maximal temperature increases observed *in vitro* and predicted *in silico*. As a few replicates have been done, experimental results exhibit an important variability. More experiments should be carried to "smooth" the results. Thus, the difference between experimental and simulation results is sometimes important. On the contrary, the dispersion of simulation results is small, but not constant, as shown in figure 6.

![Figure 5: Simulation and experiment results of temperature in *P. terraeovae* masses in various cases.](image)
Though the results are promising, in "huge aggregation" cases the model tends to underestimate the temperature in the aggregation. The existence of such a bias in predictions could be caused by two phenomena that have been neglected in the model.

![Figure 6: Standard deviation (°C) of model predictions in function of the number of individuals in the aggregation.](image)

First, this model is based on the generally accepted assumption that temperature increases in aggregations are caused by the metabolic activity of larvae during the digestion process as shown in Slone et al. (2005) and Slone and Gruner (2007). But rubbings between individuals could be responsible, for a part, of the temperature increase especially when many individuals are involved: the competition to reach the food is then harder and larvae are more active. Though this hypothesis has to be validated, experiments carried with starving larvae seem to confirm it: in large aggregations, a temperature increase can be observed, even if there is no digestive activity as shown in figure 7. Nevertheless, this effect has to be precisely quantified before it can be included in the model.

Second, in small aggregations, heat transfers with the environment are important, which is not the case in bigger aggregations. Indeed, the main heat source, the feeding larvae, is isolated from the environment by larvae that are trying to reach the food. Thus, the density of larvae should be taken into account to estimate heat transfers between an aggregation and its environment. However, as the figure 8 shows, using this model in entomological expertises should increase the accuracy of the results. Indeed, in all tested cases, the difference between simulated and measured temperatures is less than the difference between environment and measured temperatures, *i.e.*, less than the difference between the temperature used to perform expertises and the temperature felt by larvae.

![Figure 7: Temperature (°C) recorded in an aggregation of starving larvae.](image)

**CONCLUSION**

In this article, an agent-based model of thermoregulation in *P. terraeovac* masses has been presented. Though the model has been validated for this particular species, it might be used to predict the temperature increase in other/mixed species aggregations as their thermoregulation behaviours are similar. Thus, this model should increase the accuracy of entomological expertises in complex cases.

To use this thermoregulation model in real world cases, a model of human body has been developed (Veremne et al. 2008). However, as the legislation on human cadaver experiments is very strict in France, experiments are conducted on cow carrions to validate the model in real world conditions. This model has already been included in ForenSeek, a decision support system for forensic entomology². Moreover, this simulator can be used as a virtual laboratory to test hypotheses about necrophagous Diptera larvae and especially the behavioural and physiological causes of the maggot mass effect.

Beyond the application to forensic entomology, these results show that a complex thermoregulation behaviour can be produced using simple reactive agents, that do not have any knowledge of the global structure of the system. In this perspective, self-organisation and non-intentional cooperation in Diptera larva masses could be studied from a more theorectical point of vue. The simplicity and efficiency of such simple organisms is interesting to understand the formal characteristics of col-

² [http://www.foreuenseek.org](http://www.foreuenseek.org)
Figure 8: Difference (°C) between the temperatures measured in the aggregation and predicted by the model (c1) and between the temperature measured in the aggregation and the temperature of the environment (c2) in function of the number of individuals in the aggregation.

Collaboration in systems composed of numerous reactive agents (Deneubourg and Goss 1989).

REFERENCES


GENE REGULATORY NETWORKS WITH MULTIPLEXES

Gilles Bernot$^{1,2}$
Jean-Paul Comet$^1$
Zohra Khalis$^{1,2}$

$^1$13S, CNRS & University of Nice-Sophia Antipolis, 2000 route des Lucioles-Les Alarigones-Euclide-B, F-06903 Sophia Antipolis CEDEX, France
$^2$Epigenomics Project, Genopole, F-91034 Évry, France
email: {bernot, comet, khalis}@iis.unice.fr

KEYWORDS
Formal modelling, gene regulation, regulatory networks, systems biology, complex systems.

ABSTRACT

When modelling gene regulatory networks, the cornerstone of the modelling process is the search of parameter values which are consistent with the known properties of the system. These parameters drive the dynamics of the system. In this article, we give a formal definition of a slight extension of the R. Thomas’ modelling, with explicit information about cooperative, concurrent or more complex molecular interactions. It considerably decreases the number of parameters and determining parameter values becomes less time consuming, making possible the study of larger systems. We illustrate this modelling framework on the biological system regulated by Thyroid Hormones during the metamorphosis of tadpole and controlling the tail resorption.

INTRODUCTION

To study complex biological systems, formal modelling is often mandatory since the complexity of the interleaved interactions between constituents makes intuitive reasoning error prone. Numerous mathematical modelling frameworks have been proposed to model gene regulatory networks, see for example (de Jong 2002, Smolen et al. 2000, Wilkinson 2006, Fujita et al. 2004). Common approaches are quantitative, based on differential or stochastic equations, providing numerical simulations of the system. Nevertheless actual predictions often remain only qualitative because the parameter values of these systems are not precisely known. Several other modelling frameworks are based on a qualitative view, see for example boolean networks and their generalizations (Thomas 1973, Thomas et al. 1995), Petri nets (Chaouiya et al. 2004, Comet et al. 2005), hybrid modellings (Siebert and Bockmayer 2007, Ahmad et al. 2006 Cover Date: November 2007), and stochastic π-calculus (Ciochetta and Priami 2007). Each modelling framework highlights some view of models and allows one to detail or to abstract different biological aspects. We focus here on Thomas’ modelling, in which the gene regulatory system is represented by an interaction graph and a set of parameters. The interaction graph is composed, on the one hand, of nodes which abstract genes and their proteins, and on the other hand, of edges which represent the interactions between the genes. The values assigned to the parameters permit one to deduce the dynamics of the system from the interaction graph. Even in a qualitative perspective, the lack of reliable data about the system leads to a typical difficulty of the modelling approach: How to select the parameter values of the model?

For determining values of parameters, we proposed in (Bernot et al. 2004) to test the set of all possible parameterizations against temporal properties. It is finite in the case of Thomas’ modelling. This approach can be computer aided (Bernot et al. 2004) using formal temporal logics and systematic model checking. Even if the set of possible parameterizations is finite, it exponentially grows with the size of the interaction graph. Several theorems established in the Thomas’ framework considerably reduce the number of generated parameter sets, nevertheless, an entire exploration is not conceivable for large networks.

In order to reduce the time required by this exploration step, it becomes crucial to introduce in the modelling framework more biological information (when available). In this paper, we propose to take into account information about how constituents of the system act on their targets. For example (Figure 1), if two genes act positively on a common target via the formation of a complex (e.g. the transcription factor of the common target contains the complex), then it is obvious that the common target has in fact a unique predecessor (the complex instead of two genes separately) and only two possibilities (instead of four) can occur: The complex is present and the transcription can take place or the complex is not present. Indeed this idea is far from being new but it has never been formalized up to now. R. Thomas remarked that this kind of information can be taken into account in its modelling framework through the valuation of parameters, but he did not explicitly include
such information in the interaction graph (Thomas and d’Ari 1990).

![Figure 1: Example of cooperative action](image)

Here, we propose a modelling framework in which the interaction graph makes such cooperative or concurrent biological phenomenon explicit. The decreasing of the number of parameters coupled with the methodology developed in (Bernot et al. 2004), will make possible the study of larger systems.

The paper is organized as follows. We firstly define our new interaction graph: **Multiplexes** are formally defined to take into account available biological information describing the cooperation or concurrency between constituents acting on a common target. Then we define when a multiplex has an effective action on its targets, and we construct the associated dynamics. We show that Thomas’ and multiplex frameworks have the same power of expression but we illustrate, through the classical example of the lac operon, how multiplexes allow us to be more legible and terse. Lastly we illustrate multiplex modeling on the system regulating Thyroid Hormones during the metamorphosis of the tadpole and controlling the tail resorption.

**GENE REGULATORY GRAPHS WITH MULTIPLEXES**

Formal modelling frameworks for gene regulatory networks represent interactions between entities (genes, proteins, etc.) via a static graph. Then, dynamics focus on the evolution of entity expression levels and ask for more elaborated mathematical stuff with many parameters.

In our framework, we represent the static part by a directed graph composed of two types of vertices: Variables which correspond to genes and their products, and multiplexes which correspond to interactions between variables. Multiplexes abstract biological phenomena like complex forming or more elaborated phenomena. The predecessors of a multiplex are either variables or other multiplexes brought into play in the interaction; the successors are called the targets of the interaction.

**Formal Definition**

The following notation will be useful.

**Notation 1** Given a directed graph $G$ and a node $v$ of $G$, $G^{-1}(v)$ is the set of all nodes $v'$ of $G$ such that $(v', v)$ is an edge of $G$ (set of predecessors of $v$).

A multiplex is provided with a formula in a propositional logic which encodes the situations in which the interaction occurs. For example, if a complex formed with proteins $a$ and $b$ is required in cooperative action and if the complex $(a-b)$ is inactive in the presence of a protein $c$, then the corresponding formula looks like $a \land b \land \neg c$, where the symbols “$\land$” and “$\neg$” stand for “and” and “not” respectively.

**Definition 1** A gene regulatory graph with multiplexes, RG for short, is a tuple $G = (V, M, E_V, E_M)$ such that:

1. $(V \cup M, E_V \cup E_M)$ constitutes a (labelled) directed graph whose set of nodes is $V \cup M$ and set of edges is $E_V \cup E_M$, with $E_V \subset V \times \mathbb{N} \times M$ and $E_M \subset M \times (V \cup M)$.

2. $V$ and $M$ are disjoint finite sets. Nodes of $V$ are called variables and nodes of $M$ are called multiplexes. An edge $(v, s, m)$ of $E_V$ is denoted $(v \xrightarrow{m} s)$ where $s$ is called the threshold.

3. Each variable $v$ of $V$ is labelled with a positive integer $b_v$, called the bound of $v$.

4. Each multiplex $m$ of $M$ is labelled with a formula belonging to the language $L_m$ inductively defined by:
   - If $(v \xrightarrow{m} s) \in E_V$, then $v_s$ is an atom of $L_m$, and if $(m' \rightarrow m) \in E_M$ then $m'$ is an atom of $L_m$.
   - If $\phi$ and $\psi$ belong to $L_m$ then $\neg \phi$, $(\phi \land \psi)$, $(\phi \lor \psi)$ and $(\phi \Rightarrow \psi)$ belong to $L_m$.

5. All cycles of the underlying graph $(V \cup M, E_V \cup E_M)$ contain at least one node belonging to $V$.

Note: Condition 5 is necessary for the definition of dynamics (see Definition 3).

![Figure 2: Graphical conventions](image)
States and resources

A gene regulatory graph with multiplexes constitutes the static representation of the system. We have now to focus on the dynamics of the system, abstracted by the evolutions of expression levels of the variables. Let us first define the states of a system.

**Definition 2** A state of a RG $G = (V, M, E_V, E_M)$ is a map $\eta : V \rightarrow \mathbb{N}$ such that for each variable $v$ belonging to $V$, $\eta(v) \leq b_v$. $\eta(v)$ is called the expression level of $v$.

A multiplex does not have any expression level because it is a logical composition of variables at a given time. So, we consider only the expression level of all the variables at that time and from this current state it is possible to deduce if the multiplex is active or not via the interpretation of its propositional formula. According to a current state, the set of resources of a variable $a$ is the set of multiplexes which can help $a$ to express its product. More precisely a resource $r$ of a variable $a$ is a multiplex belonging to $G^{-1}(a)$ whose formula is satisfied.

**Definition 3** Given a RG $G = (V, M, E_V, E_M)$ and a state $\eta$ of $G$, the set of resources of a variable $v \in V$ for the state $\eta$ is the set of multiplexes $m$ of $G^{-1}(v)$ such that the formula $\varphi_m$ of the multiplex $m$ is satisfied. The interpretation of $\varphi_m$ in $m$ is inductively defined by:

- If $\varphi_m$ is reduced to an atom $v_s$ of $G^{-1}(m)$ then $\varphi_m$ is satisfied iff $\eta(v) \geq s$.
- If $\varphi_m$ is reduced to an atom $m' \in M$ of $G^{-1}(m)$ then $\varphi_m$ is satisfied iff $\varphi_{m'}$ of $m'$ is satisfied.
- If $\varphi_m \equiv \psi_1 \land \psi_2$ then $\varphi_m$ is satisfied if $\psi_1$ and $\psi_2$ are satisfied; and we proceed similarly for all other connectives.

We note $\rho(v, \eta)$ the set of resources of $v$ for the state $\eta$.

This definition is actually inductive because RG never contain a cycle of multiplex (item 5 of Definition 1). If cycle of multiplexes were allowed then indeterminations or contradictions would be possible. For instance, consider the graph in figure 3. Suppose that the expression level of $a$ is greater or equal to the threshold $s$:

- If the formula of $m'$ is assumed to be satisfied, then the formula of $m$ must be satisfied and so the formula of $m'$ cannot be satisfied. So, we get an inconsistency.
- If the formula of $m'$ is assumed to be unsatisfied, then the formula of $m$ must be unsatisfied and so the formula of $m'$ must be satisfied. So, whatever we assume, we always get an inconsistency.

Let us consider now, the graph in figure 3 where the formula associated with $m'$ is $m$ instead of $\neg m$. Suppose again that the expression level of $a$ is greater or equal to the threshold $s$. Then, the two interpretations of $m'$ are consistent and compatible with the current state. There is an indetermination which is similar to the notion of schizophrenic cycles of (Tardieu and de Simone 2004).

![Figure 3: Cycle of multiplexes](image)

To avoid these inconsistencies and indeterminations, cycles of multiplexes are not allowed. This motivates the item 5 of Definition 1.

Graphical conventions

Edges of interaction graphs have no sign but negative actions are taken into account through multiplexes with the operator $\neg$. For example, in Figure 2 the multiplex $m$ represents an inhibition (the complex $a-b$ inhibits $b$ and $d$ via $m'$). In figure 2, we also see that in multiplex formulas the variables are indexed by their thresholds. This is useful when a given variable acts on a multiplex at several thresholds. The multiplex $m'$ means that the expression level of $c$ must be both greater than 2 and lower than 5 in order to participate to the induction of $d$.

In addition to these standard graphical conventions, we allow “light” additional graphical notation abuse:

- If a variable is an input of a multiplex with only one threshold, we then allow to omit the threshold in the formula. For example, in Figure 2, the formula of multiplex $m$ can be simply written as “$\neg(a \land b)$.” Of course, this light form is not possible for $m'$.
- Multiplexes with a formula reduced to a unique atom can be removed from the diagram. In figure 4a, removing the multiplex $m$ allows us to retrieve the usual diagrammatic convention of R. Thomas for activations.
- Similarly, in figure 4b, we retrieve usual inhibitions, either by adding the minus sign, or by using the “inhibition arrow” usual in biology.

GENE NETWORKS WITH MULTIPLEXES

We call network a graph associated with the parameters which determine the dynamics.

**Definition 4** A gene regulatory network with multiplexes (RN) is a couple $(G, K)$ where
Definition 5 Let \( N = (G, \mathcal{K}) \) be a RN, and let \( \eta \) be a state of \( G \). A state \( \eta' \) of \( G \) is a successor of the state \( \eta \) if and only if:

- There exists a variable \( u \) such that \( \eta'(u) = \eta(u) + d(u) \) and \( d(u) \neq 0 \)
- For any other variable \( v \neq u \) we have \( \eta'(v) = \eta(v) \)

In each state transition, at most one variable is modified; this procedure is called asynchronous update in Thomas’ framework.

Definition 6 The asynchronous state graph of a RN \( N = (G, \mathcal{K}) \) is the graph \( S \) defined by:

- The set of vertices of \( S \) is the set of possible states of \( G \) (isomorphic to the Cartesian product \( \prod_{v \in V} [0, b_v] \)).
- The set of edges of \( S \) is the set of couples \( (\eta, \eta') \) such that \( \eta' \) is a successor of \( \eta \).

RELATIVE TERSENESS WITH RESPECT TO THE CLASSICAL FRAMEWORK

Obviously our framework with multiplexes embeds the classical Thomas’ framework (Thomas 1991) as it is sufficient to translate an activation (resp. an inhibition) with a multiplex whose formula is reduced to the input variable (resp. its negation), see Figure 4. Conversely, a non atomic formula in a multiplex obviously corresponds to a constraint on the parameters (Thomas and d’Ari 1990) following an induction similar to the one of Definition 3.

Our conviction is that this kind of knowledge is a static knowledge and consequently it should be present in the interaction graph (formulas in multiplexes). When we know, for biological reasons, the nature of combined influences, this information should be included in the model as soon as possible because it considerably reduces the number of possible parameters, as shown in the example below. Of course, the nature of combined influences is not always a priori known and, in this case, according to our formalism, variables have then several inputs in the regulatory graph.

Example of lactose operon.
The cell needs carbon. Carbon is preferably obtained from glucose via a given catalytic pathway. When glucose is absent, lactose is used via an alternative catalytic pathway.

Lactose operon in E.coli is the first genetic regulatory system elucidated, by François Jacob and Jacques Monod (Jacob and Monod 1961). The induction of this system requires two conditions: Absence of glucose and presence of lactose.

An operon is a set of contiguous genes whose transcription is controlled simultaneously by a unique transcription factor. This transcription factor has an affinity with
a DNA area at the beginning of the operon, called operator and denoted O.
The lactose operon is formed by three genes denoted by Z, Y and A. The genes Z, Y and A produce respectively the enzymes β-galactosidase, permease and thiogalactoside transacetylase.

When glucose is absent, the alternative pathway is controlled as follows:

- **CAP** (Catabolite gene Activator Protein) forms a complex with cAMP (cyclic Adenosine Monophosphate), and binds to DNA to increase the transcription of the operon. This is a positive regulation.

- The transcription of the operon is possible only if the DNA area O is free. The regulatory protein lacI binds to O, this is a negative regulation. However, when lactose is present, a lactose isomer binds to lacI and lacI loses its affinity for O. So the operator O becomes free.

When glucose is present, the alternative pathway is inhibited as follows: Glucose inhibits indirectly cAMP and leads to the absence of complex CAP-cAMP. Consequently, there is no transcription even if lacI is present.

![Figure 5: Lactose operon metabolism graph with and without multiplexes.](image)

In Figure 5, the interaction graph of the lactose operon is represented in the multiplex framework (the left part of the figure) and in the classical Thomas’ framework (the right part of the figure). The first advantage of the multiplex framework is its legibility: the left hand side of the figure is, to some extent, more legible than the textual description given before. On the contrary, the right hand side of the figure cannot be understood without the textual description.

The second advantage of the multiplex framework is methodological. When we try to elucidate a biological system using Thomas’ framework, we do not know the values of the parameters: The $K_{v,\text{act}}$ have to be inferred from in vivo behaviours. Consequently, models with a small number of parameters allow us to rapidly converge towards the elucidation of the studied biological system. On the contrary, models with large numbers of parameters can be so heavy to manipulate that they obstruct the discovery process. On this small lactose operon example, the total number of parameters according to the multiplex approach is 12, while the total number of parameters according to the classical approach is 51. Putting as much static information as possible explicitly in the graph (instead of putting it later manually in the dynamics) considerably reduces the complexity of the modelling methodology. Indeed, formalizing cooperative actions of several variables on the same target via multiplexes enables one to merge into a single multiplex the different acting resources.

The knowledge formalised into multiplexes can lead to reduce even more the number of useful parameters. In figure 6, multiplexes $m_1$ and $m_2$ cannot be satisfied for the same state: $m_1$ is active only if expression level of a is strictly less than 2 whereas $m_2$ is active when expression level of a is greater or equal to 2. Among the set of formal parameters $K = \{k_{v,\text{act}}(m_1), k_{v,\text{act}}(m_2), k_{c,(m_1,m_2)}\}$, $k_{c,(m_1,m_2)}$. More generally, when two multiplexes having the same target $v$ have two mutually exclusive formulas $\phi_1$ and $\phi_2$, all parameters of the form $K_{v,\text{act}}(m_1,m_2)$ can be ignored and the number of relevant parameters is reduced.

![Figure 6: Example of RG which contains mutually exclusive formulas](image)

**TH-DEPENDENT REGULATION OF TADPOLE TAIL RESORPTION**

Anuran metamorphosis is an intense period of postembryonic development that is characterized by an extensive remodelling of the tadpole into a juvenile frog: (Shi 2000, Veldhoen et al. 2002). The metamorphosis of the tadpole starts with limb growth and differentiation and ends with tail and gill resorption. We illustrate our modelling framework on a simplified model of the system regulating the tail resorption. We show that multiplexes reduce the number of parameters while making more explicit biological regulatory phenomena. After having defined variables and interactions of this regulatory system, we translate the knowledge on regulatory phenomena into multiplexes and compare the numbers of models to consider in both classical and multiplex modelling frameworks.
Variables

Two types of thyroid hormones (TH) have to be taken into account: The tri-iodothyronine (T3) and the thyroxine (T4) (Brown and Cai 2007). These hormones are regulated by two enzymes: Deiodinase of type 2 (D2) and deiodinase of type 3 (D3) which have a different role in the system. We also consider activators of D2 (intermediate genes, IG for short), early genes (EG) and late genes (LG) which are both responsible of tail resorption via apoptosis (notice that LG are expressed after EG). Finally, the nuclear thyroid hormone receptor of T3 isoform β, denoted TRβ, is explicitly represented even if it is a early gene because it has an important role (Wang and Brown 1993, Troncale et al. 2007).

Regulations

Each variable has two expression levels 0 and 1 except T3 which has four expression levels. Thus, only thresholds of edges outgoing from T3 have to be specified since other thresholds can only have the value 1.

The gene expressing D3 is a direct response gene of TH so D3 expression level increases with that of T3 at the threshold 1. D2 is regulated by TH through intermediate genes (IG) so D2 expression level increases with that of T3. However, the response time is rather long (some days) due to the cascade: T3 activates IG at the threshold 1 and IG activate D2. The transcription of EG (including TRβ) is activated when T3 reaches the threshold 2. Similarly, the transcription of LG is activated when T3 reaches the level 3. T4 is transformed into T3 by losing an iodine at a specific place: This is represented by an activation of T4 on T3. The deiodinase D2 activates T3 (via the transformation of T4 into T3) whereas D3 inhibits T3 and T4 (via the transformation of TH into inactive forms of TH). A gene of EG is probably responsible for the D3 inhibition: (Huang et al. 2001). TRβ (combined with TH) is supposed to be a transcription factor of LG (Wang and Brown 1993) whose products degrade D2 and TRβ. A first regulatory graph according to the classical approach of R. Thomas has been established from this biological information (Figure 7).

Multiplexes

In this system, D2 is a catalyst which allows the transformation of T4 into T3: (Huang et al. 2001). T4 loses an iodine under the action of D2. So, the synthesis of T3 from T4 by D2 does not consume D2. This catalysis is modelled, in the figure 8, by the multiplex named catalysis which is labelled by the formula \( T4_1 \land D2_1 \) acting on T3. Similarly, D3 catalyses the transformations of T3 and T4 into inactive forms of TH (which are not represented in this model). The catalysis of D2 on T4 does consume T4 but less than the quantity brought by the sanguine flux. The action of D3 on T4 and T3 decreases significantly their concentrations. These catalyses are then modelled by multiplexes labelled by the formulas \( \neg(T3_1 \land D3_1) \) and \( \neg(T4_1 \land D3_1) \) acting respectively on T3 and T4. Notice that these inhibitions are represented by the negation operator in the formulas. TH bind with its nuclear receptor TRβ to form complexes TH/TR which are transcription factors of the genes responsible for cellular death. We transcript this information into the multiplex labelled by the formula \( T3_3 \land TR_1 \) which acts on LG. Figure 8 represents the RG with multiplexes deduced from the previous information.

Notice that Figure 8 is not really more complex than Figure 7, but it encodes entirely all the considerations “tediously” developed in this section.

Parameters

In both modelling frameworks (with or without multiplexes), we have to give a value to each parameter in
order to deduce the dynamics of the system. Because parameter values are not a priori known this leads us to consider an enormous number of parameterizations. Indeed, each variable $v$ admits 2$^n$ parameters of the form $k_{v,\omega}$ where $n$ is the in-degree of $v$ in $G (\omega \subseteq G^{-1}(v))$. Each of these parameters can take $b_v$ different values where $b_v$ is the bound of $v$. The number of parameterizations is then given by $\prod_{v \in V} b_v^{2^n}$ where $n$ is the in-degree of $v$. For the TH-dependent regulation of the tadpole tail resorption in Thomas’ framework, the number of parameterizations is on the order of $2.7 \times 10^{11}$ whereas in our multiplex framework, the number of parameterizations is on the order of $2.6 \times 10^3$. For instance, in Thomas’ framework, the variable $T3$ has 2$^3$ parameters, generating $2^3 (65536)$ different parameter settings while in our framework, $T3$ has 2$^2$ parameters, generating $2^2 (256)$ different parameter settings. The difference resides in the addition of the multiplex catalysis, which reduces the number of inward edges to $T3$ and so the number of possible parameter settings. Consequently, taking into account information about cooperation between variables (through multiplexes) leads to a significant decreasing of the number of possible models: Here, the set of possible models is cut down by a factor of 1000.

The software SMBionet-3.0 (Richard 2006) has been designed to facilitate the modelling process of genetic regulatory systems. It allows one to select models of given RQ according to their temporal properties. It takes as input a RQ and a formula in temporal logic expressing the known or hypothetical temporal properties of the system. It gives as output all the models satisfying the formula. We used SMBionet-3.0 to exhibit models which present characteristic variations of TH concentrations, observed during the metamorphosis of the tadpole. Under the Snoussi’s hypotheses (see section Biological Regulatory Networks with multiplexes) and for a given logical formula, all possible parameter settings in our framework have been explored in approximately 20 minutes whereas all possible parameter settings in Thomas’ framework have been explored in 54 hours. Notice that the ratio between both time is less than 1000 because SMBionet-3.0 optimizes the exploration of the model set.

CONCLUSION

We rigorously introduced propositional logic elements in the R. Thomas’ framework in order to take into account available information concerning the cooperation or concurrency between genes or genes products acting on the same targets. This idea is rather natural: R. Thomas introduced in (Thomas 1991) a notation that allows the representation of several actions of a unique gene on another one. Moreover, dozens of articles can be cited which use similar ideas in different frameworks (Albert and Othmer 2003), (Klaunt et al. 2006), etc.. Up to our knowledge, our contribution is the first one which rigorously formalizes this more elaborated framework.

The introduction of multiplexes makes models terser because this framework allows the gathering of edges into a single multiplex.

The major advantage of multiplex modelling is methodological: It reduces the number of parameters by formalizing additional biological information. So, the step which searches parameter values consistent with known or hypothetical properties of the system is significantly improved. These advantages open perspectives to study larger gene regulatory networks, as shown on the tadpole example.

Another advantage of multiplexes is to facilitate manipulations of networks. For example, we may develop graph folding methods in order to reduce the number of variables, at the price of possibly long formulas in multiplexes. However the role of some variables in a path is essentially to delay the global process. Consequently to improve the biological usefulness of such abstractions, it seems necessary to take delays into account. One of our future works will be to introduce delays in multiplexes.

ACKNOWLEDGEMENTS

This work was supported by grants from Région Ile-de-France. The authors thank to the members of the observability working group of Genopole for helpful discussions on the Tadpole tail resorption model, especially N. Pollet. We also thank A. Richard for critical reading of the manuscript.

REFERENCES

Ahmad J.; Bernot G.; Comet J.P.; Lime D. and Roux O., 2006 Cover Date: November 2007). Hybrid Modelling and Dynamical Analysis of Gene Regulatory Networks with Delays. ComPlexUs, 3, no. 4, 231–251.


In *ICATPN 2004*, Springer-Verlag, LNCS 3099, 137–156.


FLUID
FLOW
SIMULATION
WATER HAMMER IN NUCLEAR INSTALLATIONS. CASE STUDY IN FEED WATER SYSTEM.

Ilie Prisecaru and Daniel Dupleac
Power Plant Engineering Faculty
Politehnica University of Bucharest
313 Splaiul Independentei, 060042, sector 6,
Bucharest, Romania
E-mail: prise@cen.pub.ro,
danieldu@cen.pub.ro

Niță Iulian Pavel
Subsidiary of Engineering and Technology for Nuclear Projects
Magurele - Ilfov
409 Atomistilor Street, POB: 5204-MG-4
nitai@router.citon.ro

KEYWORDS: NUCLEAR ENGINEERING, WATER HAMMER, TRANSIENT ANALYSIS.

ABSTRACT

From transients that occur during plant operation the worst that can happen are those generated by sharp closing/opening of isolating valves. One of these effects, called the water hammer is due to a sudden raise of pressure at values even ten times higher than the nominal values.

The hydraulic hammer and other adverse effects were largely discussed in the literature available for operators and designers of the installations where this phenomenon occurs.

There are dedicated computational codes which evaluate diverse technical parameters which characterize this phenomenon.

It is known that there are some operational regimes of nuclear facilities for which at present there are no corresponding codes able to describe all the technical aspects of the abnormal operation. Among these one can mention the critical two-phase flow and the amortizing response of the pipes to sharp pressure variations.

The authors of the available codes recommend that the computational results to be checked carefully on known data from the literature.

In this paper after a selective exposition of theoretical fundamentals of the problem, a computational calculation is presented as obtained by applying the dedicated calculation code PIPENET (Sunrise System Limited, Cambridge, Great Britain). The PIPENET calculation was compared with a standard computational calculation based on theoretical correlations. We evaluated the differences between those two computational methods in order to reveal the capabilities of the computational codes in solving the hydraulic hammering problems.

In a first stage we obtained the elastic characteristics of the pipe where the phenomenon of hydraulic hammering takes place and the descriptions by differential equations of this physical phenomenon.

In the second part of the paper we performed a complete system analysis of the water hammer effect due to a fail close of the all four level control valves of the Steam Generators.

We compared the highest pressure ever registered with the design pressure of the system. We observed that the design pressure was not over-passed.

We concluded that the system is well designed to cope with the water hammer effect due to the anomalous failure considered.

INTRODUCTION

This paper presents an approach to water hammer phenomena that occurs in nuclear and non nuclear installations. We made in the beginning an introduction to water hammer theory - then a well know application of water in a forced pipe. This was done in order to observe - calibrate the PIPENET response compared with a known situation that occurs in hydraulic installations. At the end we made an application on a transient regime in a feedwater system. The regime is initiated by close of all level control valves of steam generators. Analyses are made in two cases: in the first case the head limiter of main pumps are unavailable; in the second case the head limiter of main pumps are available. We compared the peak pressure from the system with design pressure.

WATER-HAMMER - THEORY

The standard theory of water hammer is based on two fundamental correlations:
- Computation of fluid pressure wave
- Computation of fluid speed.
In order to obtain this correlations we start from second Newton law:
\[ m\ddot{a} = \sum F \]  
(1)

The trace on Ox ax of this is:
\[ \rho \frac{\pi D^2}{4} \frac{\partial V}{\partial t} = \frac{\pi D^2}{4} \frac{\partial p}{\partial t} \frac{dx}{dx} \]  
(2)

Simplifying and taking in consideration of
\[ p = \gamma v = \rho g v, \]
\[ \frac{\partial V}{\partial t} = g \frac{\partial y}{\partial x} \]  
(3)

This is cinematic correlation, the first correlation between \( V(x,t) \) and \( y(x,t) \).

The volume variation during water hammer is:
\[ dV_{tot} = dV_{di} - dV_{compr} \]  
(4)

\[ \frac{\partial V}{\partial x} = g \frac{\partial y}{\partial t} \]  
(7)

Where
\[ \frac{g}{a} = \left( \frac{1}{\epsilon} + \frac{1}{E} \frac{D}{\delta} \right)^{-1} \]  
(8)

The water hammer phenomena respects the vibrant cord correlation of
\[ \frac{\partial^2 \gamma}{\partial x^2} = \frac{1}{a} \frac{\partial^2 \gamma}{\partial t^2} \]  
(9)

And partial derivation equation (3) and (7):
\[ \frac{\partial V}{\partial t} = g \frac{\partial y}{\partial x} \]  
(10)

Figure 1. Volume Variation During Water Hammer Phenomena

Total volume variation is given by:
\[ dV_{tot} = (Q_{in} - Q_{out}) dt = \frac{\pi D^2}{4} \left[ V - \frac{\partial V}{\partial x} dx \right] dt = \]  
\[ \frac{\pi D^2}{4} \frac{\partial V}{\partial x} dx \]  
and it represents total volume increase.

Volume rise is given by the correlation
\[ dV_{\text{di}} = \frac{\pi D^2}{4} \frac{\partial p}{\partial t} dt dx \]  
(5)

Volume contraction due to pressure rise is:
\[ dV_{\text{compr}} = -\frac{V dp}{\epsilon} = -\frac{\pi D^2}{4\epsilon} dx \frac{\partial p}{\partial t} dt \]  
(6)

The minus sign indicates that at a pressure rise we will have a decrease of the volume.

Continuity equation is
\[ \frac{\pi D^2}{4} \frac{\partial V}{\partial x} dx dt = \frac{\pi D^2}{4\epsilon} \frac{\partial p}{\partial t} dt + \frac{\pi D^2}{4\epsilon} dx \frac{\partial p}{\partial t} dt \]  
This can be simplified as:

\[ \frac{\partial V}{\partial x} = g \frac{\partial y}{\partial t} \]  
(7)

Through integration by replacing variables (Euler and D’Alambert) with change of variables
\[ \zeta = t - \frac{x}{a}; \quad \eta = t + \frac{x}{a} \]  
(11)

We obtain the canonic for equation (9)
\[ \frac{\partial^2 \gamma}{\partial \zeta \partial \eta} = 0; \text{ or } \frac{\partial \gamma}{\partial \zeta} \left( \frac{\partial y}{\partial \eta} \right) = 0; \]

\[ y - y_0 = \int \phi(\eta) d\eta + F(\zeta) = \]  
\[ = -f(\eta) + F(\zeta) \]  
(12)

Going back to initial variable we obtain and deriving by variable x Allievi correlations:
\[ y = y_0 + F(\frac{t - \frac{x}{a}}{a}) - f(\frac{t + \frac{x}{a}}{a}) \]  
(13)

\[ V = V_0 - \frac{g}{a} \left[ F\left(\frac{t - \frac{x}{a}}{a}\right) - f\left(\frac{t + \frac{x}{a}}{a}\right) \right] \]  
(14)

where \( y \) is pressure value, \( y_0 \) - initial pressure value, \( t \) - time, \( a \) - wave speed in fluid, \( g \) - gravitational acceleration, \( F \) - direct function of water hammer, \( f \) - indirect function (reflected - inverse) of water hammer.

The maximal overpressure is calculated before indirect wave is produced therefore in this situation of \( f \equiv 0 \) and \( 0 < t < \frac{2L}{a} \) where \( L \)
represents the length of the pipe from suction area to closing valve. If the valves is closing in interval \( \left( 0 ..., \frac{2L}{a} \right) \) results that at the point \( x = 0 \) and
\[ L = 0 \] the maximal rise of pressure is given by Jukovik correlation:
\[ \Delta p = \rho a V_0 \]  \hspace{1cm} (15)

where \( \rho \) is bulk density of the fluid, \( a \) - wave speed in the fluid, \( V_0 \) - fluid speed not perturbed by valve closing process.

Usual calculation is consisted of determination of functions \( F \) and \( f \) at the first step of calculation. These values obtained at the first step are used as initial data for the next iteration step.

After the computation of the first is made we will get other values for the analytical functions \( F \) and \( f \) as well for pressure and speed. These values are initial values for next iteration steps.

The iteration time step should be enough small in order to assure mathematical stability of numerical method utilized and in the same time should be correlated with closing valve stroking time.

Practical calculation method eliminates the function \( f \) based on the fact that it is equal with the value of function \( F \) at the previous computational step so that the computation is simplified to determining the function \( F \).

This computation method does not take in consideration fluid viscosity and so there is no amortization in pressure oscillations obtained by this method. Pressure waves obtained in this method are propagated infinitely. This inconvenient can be overcome by taking in consideration fluid viscosity. This problem is solved in PIPENET computational method.

In the next figure we illustrated the pressure resulted in the standard problem of water hammer - closing of a valve from a forced pipe. The results were compared with known results from technical books. The results were under 2\% difference - the differences are mainly generated by the fact that PIPENET program take in consideration the damp due to fluid viscosity and the difference between speed of sound and pressure wave in the fluid.

The work case:
- Speed of fluid: 2.5 m/s;
- Head rise in forced pipe: 150 m;
- Pipe diameter: 1.600 m;
- Pressure drop is located in valve;
- Closing time of valve: 3.2 seconds;
- Transients started at 5 seconds from the start of the analysis

The peak pressure was about 19.5 bar (g) obtained 1 second after the transient started.

**APPLICATION OF WATER HAMMER ON FEED WATER SYSTEM**

We took in consideration the feed water system. The limits of the system were:
- Upstream: Deaerator
- Downstream: Steam Generators

Transient are generated by closing the all four steam generator level control valves. Stroking time of these valves is considered to be 20 seconds. The analyses is started from nominal operating condition of feed water system - 2 main pumps in operating and one stand by pump with heating line opened.

The system flow diagram is presented in figure 4. The computational diagram is shown in the figure 5.

The computation took in consideration two cases:
- the head limiter of feed water pump unavailable;
- the head limiter of feed water pump available;
The analysis purpose was to observe the peak pressure generated by the water hammer transient and to compare it with the design pressure of the system.

Figure 6 and 7 present the maximum pressure from the system. These values are obtained at the inlet point of the level control valves.

The main difference from these two analyses was the fact that in the second one of the pumps are stopped about 14.5 seconds from the beginning of water hammer transient due to high pressure in discharge pipe. The discharge pressure of the pumps starts to decrease function of angular impulse of the pump. In order to supply this value we had to make an equivalent impulse of main pump, booster pump and their motors.

The peak pressures in analyses were:
- about 100 bar (g) for the first case;
- about 90.5 bar (g) for the second case

Comparing the design pressure of the system with computational results we conclude that the head limiter is a very important component of the system.
CONCLUSION

In this paper we tested the capabilities of computational program PIPENET to solve water hammer problems that occur in pipe systems.

We started with solving a well know problem from technical literature and comparing the results between numerical calculation with numerical methods (Alliev correlations) and computational program (PIPNET). The differences between differences between methods were insignificant and were generated by the fact that PIPNET take in consideration amortizations caused by fluid viscosity.

After that we made a transient analysis on a complex system: feed water system. These analyses revealed the importance of maximal head protection in order to protect the system against the pressure peak.

We studied nine transient regimes derived from:
- valve closing time from 1 to 15 and 24 seconds
- overhead protection of the main feed water pumps available and unavailable
- warm up line of standby pump available or not
- for all 9 regimes we made parametrical analysis for speed of sound, and pipe thickness.

From the results we observed that this protection reduced the peak pressure from 100 bar (g) to about 90.5 bar (g) meaning lowering the maximum pressure under the projecting pressure of the system.

On the other hand we tested the parametric analysis on the response of the system to water hammer phenomena. We varied several parameters as speed of sound, speed of water hammer propagation, pipe thickness, close/open time of the level control valves, availability - unavailability of maximum head of the main pumps. This complex analysis has given us the importance of having a system to check input data for consistency and accuracy and validating of stationary initial regime in order to obtain an accurate system response.

Taking in consideration the very high pressure peak resulting during water hammer phenomena, we conclude that all nuclear systems should be tested to water hammer analyses in order to obtain the maximum pressure that can occur during abnormal operating of a nuclear power plant.

On the other hand there are measures to be taken in order to limit the effect of water hammer phenomena: inferior limiting of control and operating valves stroke time, not oversize the thickness of the pipe in order to limit the sound speed in the system.

This paper shows importance of transient analysis of a system during design and operating of a system.

BIBLIOGRAPHY

[1] **Juliea FLOREA, Gh. ZIDARU** Bazele hidraulicii - culegere de probleme- -EDP. București 1969

[2] **Aureliu LECA, Ilie PRISECARU** Conducete pentru agenți termici - Ed. tehnica București 1986


[5] *** Feed water system - design manual


AUTHOR BIBLIOGRAFY:

**ILIE PRISECARU** was born in 1947 in Bucharest, Romania. He is the Head of the Nuclear Power Department at “Politehnica” University of Bucharest, Faculty of Power Plant Engineering, member of Romanian Nuclear Energy Association board and expert for the Romanian Nuclear Regulatory Authority. Author or co-author of 17 books and more than 150 papers in journals or conference sessions, he has been working in the energy field related to nuclear and thermal power plant engineering and modeling of nuclear power plant processes and equipment operation.

**IULIAN PAVEL NITA**: was born in Bucharest, graduate Polytechnic University of Bucharest in 1999, Nuclear Power Department. After that he worked at Subsidiary of Technology and Engineering for Nuclear project being involved mainly in thermal hydraulic analyses. He also graduated Nuclear Safety and Radioprotection Master at Polytechnic University of Bucharest in 2000.

**DANIEL DUPLEAC** was born in Bucharest, Romania in 1963. He studied nuclear power plant engineering at “Politehnica” University of Bucharest and obtained his master degrees in 1987 and Doctor of Science Diploma in 1998. He worked for the Cernavoda Nuclear Power Plant and Center of Engineering and Technologies for Nuclear Projects. Since 1990 he lectures at Nuclear Power Department at “Politehnica” University of Bucharest, Faculty of Power Plant Engineering and conducts research in: thermal hydraulic analysis of CANDU reactor and systems, safety analysis of nuclear reactors, modeling of two-phase flow in fuel channel, modeling of CANDU systems.
MATHEMATICAL MODELS FOR HYDRAULIC CALCULATION AND OPTIMIZATION OF COMMUNAL WATERNETS

Lucyna Bogdan and Jan Studzinski
Polish Academy of Sciences
Systems Research Institute
Newelska 6
01-447 Warsaw, Poland
E-mail: studzins@ibspan.waw.pl

KEYWORDS
Mathematical modeling, optimization, calibration, waternet hydraulic models.

ABSTRACT
In the new EU countries and among them in Poland the revitalization of old waterworks infrastructure is one of the most important tasks at the current time. To do it in optimal way the complex computerization of the waterworks is needed what implies the branch oriented development of numerical maps, monitoring systems and hydraulic models. In the paper the hydraulic waternet models developed in the Systems Research Institute and introduced into the waterworks in the Polish waterworks in Rzeszow are presented. There are two hydraulic models implemented called OHIO and MOSUW and based on Cross method and Newton method, respectively, for solving the model equations.

INTRODUCTION
In the Systems Research Institute of Polish Academy of Sciences (IBS PAN) an information system for the management of municipal waternets has been developed and introduced into a Polish waterworks (Studzinski, 2007). The main modules of the system are the branch numerical map based on the GIS system called G/Technology and made by Intergraph, the monitoring system based on the software Procon made by Schulz-Infoprod, and two waternet hydraulic models developed in IBS PAN. In the modeling programs some optimization algorithms for waternet planning and waternet control are included. These algorithms and also the algorithms to calculate the model equations are different in the both modeling programs. These differences were the cause to develop two separated hydraulic models that could be used then to compare the advantages and disadvantages of different calculation approaches.

THE WARENET MODEL EQUATIONS
In Figure 1 an example of a waternet consisted of s=3 loops, k=6 nodes and r=7 pipes is shown. With a waternet model the values of k water pressures in the nodes and of r waterflows in the pipes are to calculate and the number of the unknown parameters is n=k+r. Between n, k and s exists the relation r=k+s−1 what results in the formula n=2k+s−1=14. To solve the problem we need n model equations and they are formulated using the Kirchhoff’s laws known from the electrical engineering and also the Bernoullie’s principle known from the flow mechanics.

![Figure 1: Exemplary water net graph.](image)

From the I Kirchhoff’s law k equations for the flow balances in k nodes, linear regarding the flows $Q_i$ or $q_i$ , result:

- Node 1: \[ Q_1 - q_1 - q_2 = 0 \]
- Node 2: \[ q_2 - q_3 - q_4 = 0 \]
- Node 3: \[ q_4 - q_5 - Q_3 = 0 \]
- Node 4: \[ q_5 - q_6 - Q_4 = 0 \]
- Node 5: \[ q_3 - q_6 - q_7 = 0 \]
- Node 6: \[ q_1 + q_7 - Q_6 = 0 \]

From the II Kirchhoff’s law s equations for the pressure balances in s loops, nonlinear regarding the flows, result (h_i – pressure reduction / pressure loss along the pipe):

- Loop 1: \[ h_{1} - h_{7} - h_{3} - h_{2} = 0 \]
- Loop 2: \[ h_{3} - h_{6} - h_{5} - h_{4} = 0 \]

From the Bernoullie’s principle (k−1) linear equations for the pressure values $P_i$ (k−1) nodes result:

- Node i: \[ P_i = P_{i+1} - h_i \]

From (1), (2) and (3) the whole number of the model equations $g = k + s + k - 1 = 2k + s - 1$ results that is equal to the number of the model parameters.

For the introduced variables the following formulas are valid:

Darcy-Weisbach’s formula:

\[ h_i = \frac{\nu_0^2}{2gD} \]
with the friction coefficient \( \lambda \) and \( v = \frac{4q}{\pi D^2} \), and

**Bernoulli’s equation:**

\[
\frac{v^2}{2g} + \frac{P}{\gamma} + H = \text{const} \tag{5}
\]

with the water density \( \gamma \). Under the condition \( v_i = v_{i+1} \) the following equations result from (5):

\[
p_{i+1} + H_{i+1} \gamma = p_i + H_i \gamma
\]

\[
P_{i+1} = P_i
\]

Equation (6) is valid only in an ideal case without any pressure reduction while flowing the water through a pipe. Otherwise equation (6) takes the form

\[
P_{i+1} = P_i - h_i
\]

that is already shown in (3).

The friction coefficient \( \lambda \) in (4) can be described by the **Nikuradse’s formula**

\[
\lambda = \frac{1}{(2 \log \frac{D}{k} + 1.14)^2}
\]

that depends on the roughness coefficient \( k \) whose values are given in special tables relating to the materials of the waternet pipes.

The numerical computing of model equations (1), (2) and (3) can be made using the standard methods for solving nonlinear algebraic equation systems or using some specialized algorithms developed for waternet descriptions. These different approaches have been applied in the models presented farther down.

**THE WATERNET HYDRAULIC MODEL OHIO**

In the model OHIO equations (1), (2) and (3) are solved recurrently and the specialized **Cross’ algorithm** is used to solve the nonlinear equations (3). The pressure reduction \( h_i \) occurring along a pipe can be written down as follows

\[
h_i = \frac{8 \lambda l}{\pi g D^2} q^2 = sq^2 \tag{8}
\]

If the water flows in the loop pipes are calculated correctly then the conditions (3) are fulfilled. Otherwise the flows are to be changed in each loop by a value \( \Delta q_i \), where \( i = 1, 2 \) for our waternet example. Then we have from (8) the equation

\[
h_{ij} = s_y(q^2_y + 2q_y \Delta q_i + \Delta q_i^2)h_{ij}
\]

in which \( i \) means the loop number, \( j \) means the pipe number and the component \( \Delta q_i^2 \) is usually omitted as negligible. Equations (3) can be now formulated in the form

\[
\sum_{j=1}^6 h_{ij} = \sum_{j=1}^6 s_y(q^2_y + 2q_y \Delta q_i) = 0 \tag{10}
\]

with \( r_i \) – the pipes number in the loop \( i \).

From (10) the following relations to compute the flow corrections \( \Delta q_i \) result

\[
\frac{\Delta q_i}{2 \sum_j s_y q_y} = -\frac{\sum_j h_{ij} / q_y}{2 \sum_j s_y q_y} \tag{11}
\]

The flow corrections are computed in the **Cross’ algorithm** until the calculated flow changes in the pipes are smaller than the accepted calculation exactness. The algorithm converges usually to the right solution after few iterations only.

In Figures 2 and 3 two OHIO’s windows with a computed waternet are shown. After the model computing is done the calculated values of pressures and flows can be marked on the waternet graph with different colors for small, high and medium values (Figure 3).
In OHIO also a special algorithm for one-criterial optimization of the investigated waternet is implemented. It consists of the following steps:

**Step 1:** Localizing the end user nodes with the incorrect water pressure (too small or too high) on the basis of the calculation results of the waternet model.

**Step 2:** Determining on the waternet graph the tracks between the end user nodes localized and the water supplying nodes (water sources) with the biggest pressure reduction.

**Step 3:** Value reduction of the friction coefficient $\lambda$ down to the nominal value on the pipes included into the tracks determined.

**Repetition of Steps 1 and 2**

**Step 4:** Exchange of the pipes on the determined waternet tracks for the pipes with higher or smaller diameter depending on the sign of the pressure deviation in the end user nodes.

The idea of the algorithm is following: Reduction of the friction coefficient in the pipes in Step 3 corresponds to the procedure of pipes cleaning that is commonly done in the waterworks. Exchange of the pipes in Step 4 means the modernization work on the waternet. Both activities are connected with different costs and the exchange of the pipes costs more than their cleaning. Then we can see that in the algorithm also the costs of the waternet optimization are indirectly taken into account. In this way the two-stage-algorithm of one-criterial optimization (Stage 1 with Steps 1, 2, 3 and Stage 2 with Steps 1, 2, 4) used here corresponds practically to the multi-criterial one with two goal functions.

**THE WATER NET HYDRAULIC MODEL MOSUW-REH**

In the model MOSUW-REH equations (1), (2) and (3) are computed iteratively with the use of the classic Newton-Raphson algorithm for solving nonlinear algebraic equations.

In Figures 4 and 5 two MOSUW’s windows with the same waternet graph as in Figures 2 and 3 are shown. After the model computation the calculated values of pressures and flows can be marked on the waternet graph with different colors in the same way as it is done in the OHIOS model (Figure 5).

In MOSUW-REH an algorithm for multi-criterial optimization developed by Straubel (Straubel, 2007) is used. To execute it several goal functions can be formulated and usually they are as follows:

- $F(1)$ (min): maximal difference between the given and calculated pressure in the end-user nodes
- $F(2)$ (min): sum of the pressure losses in all waternet pipes
- $F(3)$ (min): maximal pumping pressure defined for the pump stations
- $F(4)$ (max): minimal velocity of the water flow in the waternet pipes
- $F(5)$ (min): entire investment costs as a result of the waternet optimization

- $F(6)$ (min): price of 1 m$^3$ of water.

As a result of the algorithm’s execution a set of Pareto-optimal solutions is received from which the waternet operator has to choose the solution that is best in his opinion.

Figure 4: MOSUW’s window with the waternet graph before the model calculation.

Figure 5: MOSUW’s window after the waternet calculation.

**WATER PRESSURE AND FLOW DISTRIBUTIONS USING THE KRIGING APPROXIMATION**

While operating a waternet it is important for the operator to know the current state of flows and pressures in the network. Using a hydraulic model he can calculate these values but the enormous number of them ordered mostly in the form of big tables makes it quite impossible to recognize quickly in which part of the waternet they are incorrect. In such cases
some improvement procedures have to be executed to avoid possible network failures or claims of the water consumers. Therefore it is useful to give the operator a tool for a quick assessment of the quality of the water network. To do it we have been used the kriging approximation to design the distributions of water pressures and flows in the form of contour line maps. On these maps the water network parts with right or bad functioning are marked with different colors (Studzinski and Bogdan, 2007).

The kriging approximation means the estimation of unknown values of a variable in some selected points of an area (calculation points) with the help of the known values of this variable collected in other points of this area (measurement points). The unknown value of the variable investigated in the calculation point \( x_o \) is estimated by the formula

\[
z(x_o) = \sum_{i=1}^{N} \lambda_i z(x_i)
\]

with the measured variable values \( z(x_i) \) in \( N \) measurement points and some weights coefficients \( \lambda_i \) which have to be calculated.

The calculation algorithm of the kriging approximation consists of 4 following steps (Bogdan and Studzinski, 2007):

**Step 1**: Calculation of the function called experimental semivariogram on the basis of the data picked up from the measurement points:

\[
\gamma(h) = \frac{1}{2n_h} \sum_{i=1}^{n_h} (z_{h+i} - z_i)^2
\]

where \( z_i, z_{h+i} \) are the variable values in the points which are outlying up to distance \( h \) from each other and \( n_h \) is the number of all pairs of these points.

**Step 2**: Modeling of the experimental semivariogram achieved using different analytical functions and with the help of methods of static optimization.

**Step 3**: Calculation of the weight coefficients \( \lambda_i \) from the following equations:

\[
\gamma(x_j, x_0) = \sum_{i=1}^{N} \lambda_i \gamma(x_j, x_i) + \mu
\]

**Step 4**: Calculation of the value of the variable investigated in the new point \( x_o \):

\[
z_o = \sum_{i=1}^{N} \lambda_i z_i
\]

where \( z_i \) are the variable values in the known measurement points.

The above algorithm has been written down in program KRIPOW (Studzinski and Bogdan, 2007), (Studzinski, 2008) that cooperates with programs OHIO and MOSUW-REH. In Figure 6 and 7 the results of KRIPOW with designed pressure and flow distributions in the water network are presented. Looking on the maps the water network operator can recognize very fast the work quality of the network. On our exemplary pictures we can state that the water network works wrong: the water flows are too slow (these parts are marked with the green and blue colors in Figure 6) and the pressure values are too high (these parts are marked with the brown and red colors in Figure 7). In real cases of waternet management the operator has to undertake in such situations some improvement procedures to decrease the pressure and to accelerate the water flows. This is because too high pressures can cause some damages to the water network and too slow water velocities are mostly responsible for creation of algae in the network pipes; this results in worsening the water quality and the most visible signs of it are bad smell and taste of the water. While undertaking these improvement procedures the operator shall analyse the data tables generated by the hydraulic model which will get him more detailed and quantitative information about the nodes with inappropriate pressure and the pipes with incorrect flows.
Figure 7: KRIPOW’s window with the waternet pressures distribution.

CONCLUSIONS

In the paper two mathematical models OHIO and MOSUW for hydraulic calculations and optimization of waternets are presented. They have been implemented to calculate the communal waternet in the Polish waterworks in Rzeszów. In program OHIO the algebraic equations set describing the model is divided in two subsets – linear and nonlinear. The nonlinear part is solved using the specialized recurrence Cross’ method. To solve the optimization task one criterial algorithm is calculated using technical goal function that concerns the pressure values in the end-user nodes of the waternet. The model solution in program MOSUW is based on the classic Newton-Raphson method of solving nonlinear algebraic equation sets. The optimization task there is formulated in the form of a multicriterial optimization algorithm with up to ten technical and economic goal functions. In both cases of the optimization task the calculations are made regarding mainly the pipe diameters in the waternet: they are to be changed in this way that the differences between the fixed and calculated pressures in the end-user nodes are minimized. In the paper also the program KRIPOW is presented. It allows to recognize very quickly the right or wrong situation in the waternet by drawing the pressure and flow distributions there with the use of the kriging approximation. The simulation of the waternet system is done with the use of OHIO or MOSUW programs by assuming the various scenarios of the load of the waternet and by fixing the pressure values in the end-user nodes.

REFERENCES


PETRI NETS SIMULATION
STOCHASTIC PETRI NETS
FROM BCMP QUEUEING NETWORKS TO GENERALIZED STOCHASTIC PETRI NETS: AN ALGORITHM AND AN EQUIVALENCE DEFINITION

Simonetta Balsamo, Andrea Marin
Dipartimento di Informatica
Università Ca' Foscari di Venezia
via Torino, 155 Venice, Italy
{balsamo,marin}@dsi.unive.it

KEYWORDS
GSPN, BCMP theorem, Coxian random variable, multiclass system

ABSTRACT
In this paper we define an algorithm that converts a BCMP queueing network (QN) with multiple classes of customers into a generalized stochastic Petri net (GSPN). Product-form property of BCMP networks is preserved by the GSPN model. The algorithm can be embedded in a hybrid formalisms modelling tool. In fact a product-form model can be partially expressed in terms of queueing network and partially in terms of GSPN. Then the algorithm determines an equivalent GSPN representation of the whole system, which can be eventually analyzed using exact techniques (taking advantages of the product-form property) or by GSPN simulators. It is worthwhile pointing out that multiple classes QNs are considered; hence their GSPN representation is not trivial, as queueing disciplines must be represented.

INTRODUCTION
In this paper we consider different formalisms to model systems for performance evaluation purposes. We consider the integration of different types of models into a unique framework, in order to take advantage of the relative merit of each of the modelling formalism involved. To this aim we consider the relation between the class of queueing network models and the generalized stochastic Petri nets. Specifically, we propose an algorithm to transform a class of queueing networks into the corresponding stochastic Petri net model. Queueing networks (QNs) are useful stochastic models for the performance evaluation of systems that consist of a set of customers which compete for a set of resources. The exact analysis of such kind of models, if possible, is usually computationally unfeasible due to the high cardinality of the set of all possible system states. Some classes of QNs have been introduced which allow a closed expression for the stationary probabilities, i.e., it can be expressed as product of functions that depend on the state and on the type of each node. The main result is known as BCMP theorem (Baskett et al. 1975) and it considers open, closed and mixed QNs that consist of multiple class of customers, probabilistic routing, Poisson arrivals and four types of service centers.

Generalized Stochastic Petri Nets (GSPNs) are stochastic models that represent systems with concurrency and synchronization and they are usually defined as models at a lower level of abstraction with respect to QNs. In fact the GSPN semantic is strictly defined in terms of transitions, places and rules of firing, that are the GSPN components. We formally introduce the GSPN definition in the next section. In the general case, GSPNs are hard to study because of the generation of their reachability sets which is an NP-hard problem. Product-form GSPNs are studied in (Coleman et al. 1996, Balbo et al. 2002) and they define the stationary state probability as the product of functions depending on the marking of each place. However, these GSPNs still do not overcome the problem of deciding whether a marking is reachable from the model initial state or not. As a consequence, this limits the applicability of the algorithms defined for product-form GSPNs (Sereni and Balbo 1997, Coleman et al. 1996). Hence the solution of such models can be derived by simulation in many practical complex cases.

In this paper we present an algorithm that transforms a multiple class BCMP queueing network into a GSPN model. The task is not obvious for at least two reasons: 1- Multiple class QNs cannot be simply associated to state machines as for single class QNs. In fact theoretical results (Baskett et al. 1975, Chandy et al. 1977) have shown that the queueing discipline in queues the performance measures in multiple class QNs. Hence the GSPN that models a specific multiple class queueing station depends on the corresponding queueing discipline, and the equivalence between the QN and GSPN models must be proved.

2- BCMP QNs allow some classes of stations to serve the customers with Coxian distributed service time. Moreover the service time distribution can be class dependent. Also in this case state machines cannot be used, and a different GSPN model has to be defined.

The proposed algorithm is based on some equivalence results presented in previous papers (Balbo et al. 2003, Balsamo and Marin 4-6 June 2007; October 23-25, 2007)
that we brie y recall in the section on related works. The proposed technique has the following properties:
- the modelling approach is modular and hierarchical. Indeed we de ne some GSPN blocks corresponding to queuing stations that can be combined into more complex systems preserving the product-form property.
- the GSPN models correspondent to the BCMP service stations can be combined with other no-BCMP product-form models. We have proved in (Balsamo and Marin 4-6 June 2007; October 23-25, 2007) that any system that holds \( M \Rightarrow M \) Muntz (1972) property can be combined with our BCMP-equivalent models preserving the product-form solution.

The algorithm is based on the following idea:
- For each BCMP queueing center type it de nes a correspondent equivalent GSPN model.
- The GSPN models obtained from the queueing center translations are connected by arcs and immediate transitions according to theQN routing probability matrix. Theoretical results proved in (Balbo et al. 2003, Bal
samo and Marin 4-6 June 2007; October 23-25, 2007) de ne the equivalence property between the GSPN and QN models, in terms of stationary state probability and average performance indices. Even if translating a QN into a GSPN increases the model complexity and reduces its readability, there are also some relevant advantages. First, the class of GSPN models is very expressive and its semantic if formally de ned at a very low level. This means that given a model de nition, the state representation can be obtained automatically. In general this does not happen when studying QNs where the system state has to be derived from a high level description of the queueing discipline. Therefore the class of GSPN models represents a suitable candidate for being the base model for a hybrid modelling tool. A second reason to use GSPN is related to the former one. We observe that product-form BCMP QNs exact analyzers or simulators usually do not allow the modeler to de ne new queuing centers with speci c discipline. For example, to the best of our knowledge, modelling tools based on QNs do not allow the modeler to introduce in the network an MSCCC service center type de ned in (Le Boudec 1986) that extends the BCMP theorem. Using GSPNs, one can de ne a model representing MSCCC discipline and then can embed it into the net. Therefore the performance indices can be derived, possibly by the product-form analysis.

MODELS IN PRODUCT FORM
In this section we introduce the formalisms that we use in the following and we brie y review the main de nitions of the QNs and the GSPNs. We will limit our description to BCMP queueing networks.

BCMP Queueing networks
A queueing network consists of a set \( C = \{c_1, \ldots, c_N\} \) of \( N \) service centers or stations. Each service center has a scheduling discipline. BCMP QNs allow the following service disciplines: First Come First Served (FCFS), Last Come First Served with preemptive resume (LCFSPR), Processor Sharing (PS) and In nite Servers (IS). In a QN, the customer enters a service center, waits in the queue for the service, gets the service, and then it either exits the network or enters another service center. Customers moves among the service centers according to routing probabilities. At a given time, every customer belongs to a class, but there can be class switchings, i.e., a customer can change its class after being served at a station. The class of the customer in uences the routing probabilities and the service time at the stations. We denote by \( R \) the number of classes. The classes are labeled by \( 1, \ldots, r, \ldots R \) and can be partitioned into chains. A chain permanently characterizes a customer. In order to simplify the notation we consider BCMP QNs with multiple chains but only one class for chain. Hence in this context the terms class and chain becomes synonymous.

In the section on supported extension of the algorithm, we show how it is possible to model class switching. We use the following notation for QNs:
- \( p_{ij}^{(c)} \) with \( 1 \leq i, j \leq N \) and \( 1 \leq c \leq R \) is the probability that a chain \( c \) customer goes to station \( j \) after being served by station \( i \).
- \( p_{0i}^{(c)} \) with \( 1 \leq i \leq N \) and \( 1 \leq c \leq R \) is the probability that a chain \( c \) customer exits the network after being served by station \( i \). Then the normalizing condition holds, i.e., \( \sum_{j=0}^{N} p_{ij}^{(c)} = 1 \), for \( 1 \leq i \leq N \) and \( 1 \leq c \leq R \).
- \( \mu_i^{(c)} \) with \( 1 \leq i \leq N \) and \( 1 \leq c \leq R \) is the mean service rate for a chain \( c \) customer at station \( i \). If the service time is Coxian distributed and \( L_i^{(c)} \) is the number of stages for chain \( c \) customers at station \( i \), then \( \mu_i^{(c)} = 1 \leq c \leq L_i^{(c)} \) denotes the mean service rate for a chain \( c \) customer at stage of service \( \ell \) of station \( i \).
- If chain \( c \) is open, i.e., external arrivals and departures from the system are allowed, then \( \lambda^{(c)} > 0 \) denotes the external arrival rate for class \( c \) customers. The arrival probability at node \( i \) and chain \( c \) is denoted by \( p_{0i}^{(c)} \). It is de ned such that \( \sum_{i=1}^{N} p_{0i}^{(c)} = 1 \), for \( 1 \leq c \leq R \). Then \( \lambda^{(c)} p_{0i}^{(c)} \) is the external arrival rate of chain \( c \) customers to station \( i \). If chain \( c \) is closed then \( p_{0i}^{(c)} = 0 \).

BCMP theorem considers four types of scheduling disciplines with some constraints. FCFS stations must have exponentially distributed service time. The service time must be chain-independent, i.e., \( \mu = \mu^{(c)} \) for \( 1 \leq c \leq R \). LCFSPR, PS and IS station types have less restrictive conditions. The service time can be Coxian distributed and the mean service rate can depend on the customer being served. Let \( n = (n_1, \ldots, n_N) \) be a vector whose components are \( \ell \)-dimension vector of vectors and where component \( n_i^{(c)} \) represents the number of class \( r \) customers at station \( i \). Then BCMP theorem (Baskett et al. 1975) states that, under stability con-
ditions, the stationary probability distribution is given by:
\[
\pi(n) = \frac{1}{G} d(n) \prod_{i=1}^{N} g_i(n_i),
\]
where \(g_i(n_i)\) is a function defined according to station \(i\) type, \(d(n)\) is a function defined for state dependent arrival rates and \(G\) is a normalizing constant. From the stationary state distribution one can derive several average performance indices.

BCMP theorem holds if the service rates depend on the state of the network. Several load-dependent service time functions have been defined in (Baskett et al. 1975). However, for the sake of clarity, we restrict study stations with a finite number of identical servers and scheduling discipline IS, PS, LCFS, PR, FCFS.

**Generalized Stochastic Petri Nets**

In this section we briefly recall the Generalized Stochastic Petri Nets (GSPN). We consider the notation for GSPN introduced in (Marsan et al. 1995). In order to allow marking dependent probabilities for solving conflicts among immediate transitions we use the techniques discussed in (Chiola et al. 1993). Let us define a Stochastic Petri Net which consists of a 8-tuple as follows:

\[
GSPN = (P, T, I(\cdot, \cdot), O(\cdot, \cdot), H(\cdot, \cdot), \Pi(\cdot), w(\cdot, \cdot), m_0)
\]

where:
- \(P = \{P_1, \ldots, P_M\}\) is the set of \(M\) places,
- \(T = \{t_1, \ldots, t_N\}\) is the set of \(N\) transitions (both immediate and timed),
- \(I(t_i, P_j) : T \times P \rightarrow N\) is the input function, \(1 \leq i \leq N, 1 \leq j \leq M,\)
- \(O(t_i, P_j) : T \times P \rightarrow N\) is the output function, \(1 \leq i \leq N, 1 \leq j \leq M,\)
- \(H(t_i, P_j) : T \times P \rightarrow N\) is the inhibition function, \(1 \leq i \leq N, 1 \leq j \leq M,\)
- \(\Pi(t_i) : T \rightarrow N\) is a function that specifies the priority of transition \(t_i, 1 \leq i \leq N,\)
- \(m \in \mathbb{N}^M\) denotes a marking or state of the net, where \(m_i\) represents the number of tokens in place \(P_i, 1 \leq i \leq N,\)
- \(w(t_i, m) : T \times \mathbb{N}^M \rightarrow \mathbb{R}\) is the function which specifies for each timed transition \(t_i\) and marking \(m\) a state dependent ring rate, and for immediate transitions a state dependent weight,
- \(m_0 \in \mathbb{N}^M\) represents the initial state of the GSPN, i.e., the numbers of tokens in each place at the initial state. We consider ordinary nets, i.e., functions \(I, O\) and \(H\) take values in \([0, 1]\).

For each transition \(t_i\) let us denote the input vector \(I(t_i)\), the output vector \(O(t_i)\) and the inhibition vector \(H(t_i)\) as follows: \(I(t_i) = (i_1, \ldots, i_M)\) where \(i_j = I(t_i, P_j)\), \(O(t_i) = (o_1, \ldots, o_M)\) where \(o_j = O(t_i, P_j)\) and \(H(t_i) = (h_1, \ldots, h_M)\) where \(h_j = H(t_i, P_j)\). Function \(\Pi(t_i)\) associates a priority to transition \(t_i\). If \(\Pi(t_i) = 0\) then \(t_i\) is a timed transition, i.e., it res after an exponentially distributed running time with mean \(1/\omega(t_i, m)\), where \(m\) is the marking of the net. If \(\Pi(t_i) > 0\) then \(t_i\) is an immediate transition and its ring time is zero. We say that transition \(t_i\) is enabled by marking \(m\) if \(m_i \geq I(t_i, P_j)\) and \(m_i < H(t_i, P_j)\) for \(i = 1, \ldots, M\) and no other transition of higher priority is enabled. We consider just two priority levels, 0 and 1. Hence when an immediate transition is enabled all the timed ones are disabled. The ring of transition \(t_i\) changes the state of the net from \(m\) to \(m - I(t_i) + O(t_i)\). The reachability set \(RS(m_0)\) of the net is defined as the set of all markings that can be reached in zero or more rings from \(m_0\). We say that marking \(m\) is tangible if it enables only timed transitions and it is vanishing otherwise. For a vanishing marking \(m\) let \(T_0\) be the set of enabled immediate transitions. Then the ring probability for any transition \(t_i \in T_0\) and any state \(m\) is denoted by \(p(t_i, m)\) and it is defined as:

\[
p(t_i, m) = \frac{w(t_i, m)}{\sum_{t_j \in T_0} w(t_j, m)}.
\]

Given a tangible marking \(m\) the transition with the lowest associated stochastic time res. Sometimes it can be useful to associate a probabilistic output vector to a transition. In this case we denote a possible output vector of transition \(t_i\) by \(O(t_i, P_j)\) and its ring probability by \(d(t_i, j)\) where \(\sum_j d(t_i, j) = 1\). Note that this is not a real extension to the model definition.

A GSPN is represented by a graph with the following conventions: timed transitions are white lined boxes, immediate transitions are black lined boxes, places are circles, if \(I(t_i, P_j) > 0\) we draw an arrow from \(P_j\) to \(t_i\) labeled with \(I(t_i, P_j)\), if \(O(t_i, P_j) > 0\) we draw an arrow from \(t_i\) to \(P_j\) labeled with \(O(t_i, P_j)\), if \(H(t_i, P_j) > 0\) we draw an circle ending line from \(P_j\) to \(t_i\) labeled with the value of \(H(t_i, P_j)\), the marking \(m\) is represented by a set of \(m_j\) circles representing the tokens in place \(P_j\) for each \(j = 1, \ldots, M\). For ordinary nets we do not use labels for the arrows.

GSPN analysis consists in nding the steady state probability for each tangible marking of the reachability set, from which one can derive other average performance indices. Some analysis techniques are presented in (Marsan et al. 1995). GSPN in product-form are studied in (Balbo et al. 2002) and they are defined as GSPNs reducible to SPNs in Coleman, Henderson et al. product-form (Coleman et al. 1996). GSPN product-form can be also identified as a special case of Boucherie product-form definition (Boucherie 1994). The product-from GSPN models that we introduce with the proposed algorithm do not belong to any of the previous product-form classes.

**RELATED WORKS**

In this section we review some previous results on relations between BCMP queueing network and GSPNs. Hybrid modelling and combining different classes of
stochastic models has been studied in literature by considering various types of models. For example in (Balbo et al. 1998) the authors illustrate a hybrid GSPN/QN modelling technique although product-form is not deeply explored. In (Bause 1993) the author studies an hybrid formalism SPN/QN. He considers a SPN with product-form (Coleman et al. 1996) and then introduces a new place type which exhibits a queueing station behavior. Then he shows that the whole system maintains the product-form property. Our approach differs from the previous ones for several reasons:
- it considers multiple class QNs in product-form.
- it combines GSPNs and QNs so it allows the modeler to use immediate transitions.
- an hybrid model is translated into a standard GSPN model. Hence existing GSPN analyzers can be used in order to simulate the net or to obtain exact results.

The definition of a GSPN model that is equivalent to a multiple-class queueing station is presented in (Balsamo and Marin 4-6 June 2007; October 23-25, 2007) and we shall now informally recall the main idea. Let us consider a multiple class FCFS station. In order to define the GSPN model, when we consider the system state, as mentioned above, we cannot just count the number of customers in the stations for every class, as this technique would ignore the queueing discipline, leading to incorrect results as discussed in (Balbo et al. 2003). For each customer class we use a place for representing the customers in queue, and a place for the customer in service. Another place stores as many tokens as the free servers are. If there is a free server and a set of waiting customers an immediate transition puts a customer in service. The problem is how to choose which customer has to get the service. In (Balsamo and Marin 4-6 June 2007) we showed that we can choose the customer to put in service probabilistically, according to the uniform distribution. This is obtained by an appropriate definition of the immediate transition weights. Service time is simply modelled by a timed transition. Figure 1 illustrates the GSPN model equivalent to a two classes FCFS station with 3 identical servers.

![Figure 1: FCFS equivalent representation by GSPN](image)

Representing an LCFSPR queueing station is more complex than FCFS, because it allows Coxian distributed service times, so we have to represent every stage of service. In the GSPN model, as soon as a token representing a new customer arrives, two things can happen: 1) a customer is preempted. This happens if there are not free servers. The customer to preempt is chosen probabilistically with uniform distribution. This is modelled by immediate transitions. 2) The customer enters in service immediately. This happens if there is at least one free server. This is modelled by immediate transitions. When a token representing a customer leaves the net for a service completion, a server becomes free if there are not preempted customers, or a preempted customers is resumed otherwise. The customer to be resumed is chosen probabilistically with uniform distribution. When a customer is preempted, the correspondent token has to be stored in an appropriate place that will identify the customer class and the stage of serviced reached.

The equivalence between the models and the QN stations is de ned and proved in (Balsamo and Marin 4-6 June 2007; October 23-25, 2007) and is based on $M \Rightarrow M$. (Muntz 1972). GSPN models can be combined with any other models holding $M \Rightarrow M$ property obtaining a product-form solution. In order to decide whether a system holds $M \Rightarrow M$ property the space state has to be built, so it cannot be considered a structural property. Deciding how this property can be translated into general structural GSPN conditions is still an open problem.

**ALGORITHM DEFINITION**

We shall now define the algorithm that converts a BCMP QN with multiple classes of customers into a product-form GSPN. Let be the set of queueing stations of the BCMP network. In the algorithm we use the following syntactical conventions for the input that is the set of parameters of the QN, according to the definition introduced in the section on product-form stochastic models:
- $P$ is the routing matrix.
- $Q = \{c_1, \ldots, c_N\}$ is the set of queueing stations, and $c_i$ is a record with the following fields: $c_i,\mu(c)$ is the single server service rate, $c_i,K$ is the number of servers, $c_i,\text{type}$ is a description of the station type. For FCFS stations, we use $c_i,\mu$ to point out that class-dependent service rate is not allowed.
- If station $i$ has a Coxian service time distribution, then we use the following notation for a customer of class $r$: $c_i,\lambda_r$ is the number of stages of the random variable, $c_i,\mu_{c(r)}$ the rate of stage $\ell$, $c_i,a_{c(r)}(\ell < L_r)$ the probability that a customer goes to stage $\ell+1$ after being served at stage $\ell$, and by $c_i,b_{c(r)}$ the probability of leaving the Coxian service after being served at stage $\ell$.
- $\lambda = (\lambda_1, \ldots, \lambda_R)$ is the vector of the arrival rates for chain $r$, $1 \leq r \leq R$. If chain $r$ is closed then $\lambda_r = 0$. Vector $K = (K_1, \ldots, K_R)$ components denote the number of customers for closed chains. $K_r = 0$ for open chains.
Let us describe the output syntactical conventions that is the de  

tion of the GSPN equivalent to the given 

QN.

- \( \mathcal{P}, \mathcal{T} \) are the sets of places and transitions, respectively. Each element of \( \mathcal{P} \) or \( \mathcal{T} \) can be labeled by a superscript (e.g. \( P_{\ell,i}^j \) is labeled by an \( I \)). Subscript letters denote some variables de ned in the algorithm. In particular letter \( r \) denotes the customer chain/class, \( \ell \) the stage of a 

Coxian random variable, \( i, j \) the correspondent service center number. For example \( P_{\ell,i}^j \) is a place de ned in the 

\( i \)-th service center translation, correspondent to the 

\( \ell \)-th stage of the \( r \) class Coxian service time. 

Timed transitions use capital \( T \). Labels \( I \) and \( O \) play a special role for places, as \( P_{\ell,i}^{I} \) represents the input-place for 

\( \ell \) class customers, and \( P_{\ell,i}^{O} \) the output place. In later 

in this section we show an example.

- \( \mathbf{m} \) is a net state and \( \mathbf{M} \) is the initial state. Vector \( \mathbf{m} \) 

consists of components whose names are derived from the 

correspondent place names. For example \( m_{\ell,i}^{S} \) is the number of 

initial tokens in place \( P_{\ell,i}^{S} \).

- The arcs are speci ed in terms on input, output, in-

hibition functions as de ned above in the section on 

product-form stochastic models. Transition priorities 

can be either 0 or 1 and they are determined by func-

tion \( \Pi \) introduced above.

- The arc weights are de ned by function \( w(t, \mathbf{m}) \) for 

each timed transition \( t \) and state \( \mathbf{m} \). For brevity we 

write just \( w(t) \). As arc weights can be state dependent, a 

symbolic function must be assigned \( w(t) \). In order to 

point out this, we use the assignment symbol \( \leftarrow \) instead of the usual :=.

- Function \( d(t, j) \) de nes the probability of the output 

vector \( O_j(t, P_j) \) for a transition \( t \) and a place \( P_j \), as 

described above in the section on product-form stochastic 

models.

Before introducing the algorithm it is worthwhile 

illustrating some notes on the translation approach. The 

algorithm rst translates every QN station into a GSPN 

(sub)model. Then it combines these GSPNs obtained by the 

erst step by connecting them through a set of imme-

diate transitions that model the QN routing. In order to 
simplify the de nition of the new combined GSPN 
in product-form, we use a standard name for input and 

output places for each GSPN (sub)model corresponding 
to a station type. This can be thought as an input 

and output interface of each GSPN submodel that simpli-
es their composition (see Figure 2). Although this 
can be a complication in the net structure, a set of 

reducible immediate transitions could be generated, the 

modularity of our algorithm results really enhanced. In 

fact, let us consider station \( i \) and suppose that \( p_{ij}^{(r)} > 0 \) 

and \( p_{ik}^{(r)} > 0 \). Using input and output interfaces we can 

represent this probabilistic routing without caring about the 

queueing discipline of stations \( i \) and \( j \) as illustrated in 

Figure 3. The main structure of the algorithm is 
simple and is shown by Algorithm 1. The main cycle of 

the algorithm considers each service center of the 

QN and executes the appropriate code block. Finally, the 

queueing network routing is modelled by ROUTING-

Block. Instructions graph(g) := \( \emptyset \), where \( g \) is a function, 

are used to initialize the function de Nations to the 

empty set, i.e. their domain is initially empty. 

FCFSBlock is de ned by Algorithm 2. It generates the 

FCFS-equivalent GSPN block described in the previous 

section. \( P_{\ell,i}^{S} \) is the place for the free servers, \( P_{\ell,i}^{I} \) 

the place for queued customers of class \( r \) (and also the input 

place), \( P_{\ell,i}^{O} \) the place for customers being served. Place 

\( P_{\ell,i}^{O} \) receives the class \( r \) customers after job completion. 

Transition \( t_{\ell,i} \) puts in service a class \( r \) customer and \( T_{r,\ell} \) 

models the service time.

Let us illustrate the LCFSPRBlock. In order to clarify 

the notation, we recall that \( t \) denotes the considered 

station, \( \ell \) the Coxian service stage, \( r \) the customer class, 

label \( Q \) denotes the queue and label \( S \) denotes the service. 

The transformation algorithm for the LCFSPRBlock 
is illustrated by Algorithm 3. Place \( P_{\ell,i}^{S} \) contains the 

tokens representing class \( r \) customers in service at stage \( \ell \) 

while place \( P_{\ell,i}^{O} \) contains the preempted ones. Transition 

\( t_{\ell,i}^{p} \) implements the preemption if there is an ar-

rived customer \( (I(t_{\ell,i}^{p}) = 1) \), there is at least a class \( r \) customer in stage \( \ell \) \( (I(t_{\ell,i}^{p}) = 1) \), there are 

no free servers \( (H(t_{\ell,i}^{p}) = 1) \). Transition \( t_{\ell,i}^{R} \) 

implements the resume of a customer. Place \( P_{\ell,i}^{R} \) stores the 
class \( r \) just arrived customers that will get the ser-

vice immediately.

The ISBlock is simple and is illustrated by Algorithm 4. 

PSBlock is similar to ISBlock so the same Algorithm 4
Algorithm 1: Main program

/* Add a place for the free servers */
\[ Q := \mathcal{P} \cup \{F_i^S\}; \]
/* Add a place for the free servers */
\[ \mathcal{P} := \mathcal{P} \cup \{F_i^T\}; \]
for each \( i \in \mathcal{C} \) do
  /* Set up the arrival places and transitions */
  \[ \mathcal{P} := \mathcal{P} \cup \{P_{r,t,i}^l, P_{r,t,i}^T\}; \]
  \[ T := T \cup \{t_{r,t,i}\}; \]
  \[ I(t_{r,t,i}^l, P_{r,t,i}^l) := 1; \]
  \[ O(t_{r,t,i}^l, P_{r,t,i}^T) := 1; \]
  \[ w(t_{r,t,i}^l) := 1; \]
  /* Set up the output places */
  \[ \mathcal{P} := \mathcal{P} \cup \{P_{r,t,i}^O\}; \]
  /* Add the other needed places and transitions for each class */
  for \( \ell := 1 \) to \( \ell = L_c \) do
    \[ \mathcal{P} := \mathcal{P} \cup \{P_{r,t,i}^Q_{r,t,i+1,\ell}, P_{r,t,i}^S_{r,t,i+1,\ell}\}; \]
    /* Add transitions which model the service time */
    \[ T := T \cup \{t_{r,t,i}\}; \]
    \[ w(T_{r,t,i}) := \mu_{c_i} \cdot \mu_{c_i} + m_{r,t,i}; \]
    \[ \Pi(T_{r,t,i}) := 0; \]
    \[ I(T_{r,t,i}, P_{r,t,i}^S) := 1; \]
  /* Add transitions modelling the preemption (label \( P \)) */
  \[ T := T \cup \{t_{r,t,i}^P\}; \]
  \[ I(t_{r,t,i}^P, P_{r,t,i}^S) := 1; \]
  \[ O(t_{r,t,i}^P, P_{r,t,i}^T) := 1; \]
  \[ w(t_{r,t,i}^P) := m_{r,t,i}; \]
  /* Add transitions modelling the resume (label \( R \)) */
  \[ T := T \cup \{t_{r,t,i}^R\}; \]
  \[ I(t_{r,t,i}^R, P_{r,t,i}^Q) := 1; \]
  \[ O(t_{r,t,i}^R, P_{r,t,i}^S) := 1; \]
  \[ w(t_{r,t,i}^R) := m_{r,t,i}; \]
end

Algorithm 2: FCFSBlock

/* Add a place for the free servers */
\[ Q := \mathcal{P} \cup \{F_i^S\}; \]
/* Add a place for the free servers */
\[ \mathcal{P} := \mathcal{P} \cup \{F_i^T\}; \]
for each \( r \in \mathcal{C} \) do
  /* Add 2 places, an immediate transition */
  and a timed transition for each class */
  \[ \mathcal{P} := \mathcal{P} \cup \{P_{r,t,i}^l, P_{r,t,i}^Q, P_{r,t,i}^O\}; \]
  \[ T := T \cup \{t_{r,t,i}\}; \]
  /* Input functions of immediate transition */
  \[ I(t_{r,t,i}, P_{r,t,i}^S) := 1; \]
  \[ I(t_{r,t,i}, P_{r,t,i}^Q) := 1; \]
  \[ I(t_{r,t,i}, P_{r,t,i}^O) := 1; \]
  /* Set immediate transitions */
  \[ O(t_{r,t,i}, P_{r,t,i}^Q) := 1; \]
  \[ O(T_{r,t,i}, P_{r,t,i}^S) := 1; \]
  \[ O(T_{r,t,i}, P_{r,t,i}^O) := 1; \]
  \[ w(t_{r,t,i}) := \mu_{c_i}; \]
  /* Set timed transition rates */
  \[ w(T_{r,t,i}) := \mu_{c_i} \cdot \mu_{c_i}; \]
  /* Transition priority */
  \[ \Pi(t_{r,t,i}) := 1; \]
  \[ \Pi(T_{r,t,i}) := 0; \]
  \[ M_{r,t,i} := c_i; K; \]
end

Algorithm 3: LCFSPRBlock

/* Add a place for the free servers */
\[ Q := \mathcal{P} \cup \{F_i^S\}; \]
/* Add a place for the free servers */
\[ \mathcal{P} := \mathcal{P} \cup \{F_i^T\}; \]
for each \( i \in \mathcal{C} \) do
  switch \( c_i, \text{type} \) do
    case FCFS
      FCFSBlock;
    end
    case LCFSPR
      LCFSPRBlock;
    end
    case IS
      ISBlock;
    end
    case PS
      PSBlock;
    end
end
/* Model the routing */
ROUTINGBlock;
/* Model arrivals and closed chains population */
CHAINSBLOCK;
applies, except for the definition of function \( w \). In fact in PS stations there is a limited number of servers, hence the servers speed must be partitioned among all the customers in the station. The weight \( w \) of transition \( T_{r,\ell,i} \) for PS station is defined as follows:

\[
W(T_{r,\ell,i}) = \frac{\min \left( \sum_{t \in c_i} \sum_{u=1}^{c_i L_t} m_{t,u,i} c_i K \right)}{\sum_{t \in c_i} \sum_{u=1}^{c_i L_t} m_{t,u,i} * c_i \mu_t^{(r)} * m_{r,\ell,i}}.
\]

Place \( P_{r,\ell,i} \) contains the class \( r \) customers at stage \( \ell \) of station \( i \). Transition \( T_{r,\ell,i} \) models the stage \( \ell \) service time and its output vector is probabilistic according to the Coxian random variable parameters.

/* Set the places set */
foreach \( r \in c_i, R \) do
P := P \cup \{P^{(r)}_{r,\ell,i}\};
for \( \ell := 1 \) to \( c_i L_r \) do
/* Add place for stage \( \ell \) of class \( r \) customers */
P := P \cup \{P_{r,\ell,i}\};
/* Add transitions modelling service time */
T := T \cup \{T_{r,\ell,i}\};
W(T_{r,\ell,i}) := m_{r,\ell,i} \mu_t^{(r)} ;
I(T_{r,\ell,i}, P_{r,\ell,i}) := 1;
O_0(T_{r,\ell,i}, P_{r,\ell+1,i}) := 1;
d(T_{r,\ell,i}, 0) := c_i a_{r,\ell};
O_1(T_{r,\ell,i}, P^{(r)}_{r,\ell}) := 1;
d(T_{r,\ell,i}, 1) := c_i b_{r,\ell};
end
Let \( P_{r,\ell} \) be an alias for \( P_{r,\ell,i} \);
end

Algorithm 4: ISBlock

In the ROUTINGBlock we define a set of transitions \( t^{(r)} \), where \( t^{(r)}_{r,i} \) models the probabilistic routing for class \( r \) customers after being served by station \( i \). The main idea has been introduced at the beginning of this section. The external arrivals are modelled by appropriate timed transition that are always enabled. In order to model a chain population it suffices to set the initial marking \( M^{r}_{r,i} \) for an arbitrary service center \( i \) equals to the chain population. This work is done by CHAINSB-Block illustrated by Algorithm 6.

**SUPPORTED EXTENSIONS**

The proposed algorithm that transforms BCMP QNs into GSPNs can support some extensions of the introduced class of BCMP QNs. In this section, for sake of brevity we just cite some extensions that can be easily supported by the algorithm with small changes.

**State dependent service rate.** BCMP theorem defines several extensions of the product-form solution to
include state dependent service rates. We can represent all the extensions whose service rates depend only on the state of the stations (i.e., we exclude the service rates depending of the state of a subnet of the QN).

**Multiple chain and multiple class.** In this work we have not considered the case of customer class switching. This has been done just to keep the notation simple. In fact by introducing some easy changes to the algorithm, with a more complex state notation we can also model multiple classes and multiple chains BCMP QNs.

**Other service station queueing disciplines.** Some extensions of BCMP theorem have been de ned to allow different queueing disciplines that lead to \( M \Rightarrow M \) product-form. If a GSPN model can represent such disciplines, then the proposed transformation algorithm from QN to GSPN can be easily modified in order to include these new station types. In fact it suffices to define a station type label and extend the switch construct of Main Program to indicate that new type of station. Then the model de nition must provide an input interface and an output interface as described in the previous section.

**EXAMPLE**

In this section we sketch an example of application of the proposed algorithm, by considering also its extensions. We apply the algorithm to the queueing network illustrated in Figure 4 (a). It is a QN with three classes of customers clustered in two chains (classes A and B, class C) and there is a switch. Classes A and B form an open chain while class C a closed one. Note that the QN has product-form solution, but it is not a BCMP QN because of the presence of a MSWCC station, i.e., a queueing discipline not considered by BCMP theorem. MSCCC discipline follows a multiple servers RANDOM discipline, but cannot serve two customers of the same class simultaneously. It is described in Le Boudec (1986) and it is proved and it holds \( M \Rightarrow M \) property. Customers of class A and B have the same stochastic behavior once they reach the servers, and they leave the system at the end of the service. Class C customers can be thought as representing a set of interior control processes whose number is given, denoted by \( K = 5 \). In order to simplify the system model we assume that all the service times are exponentially distributed. We assume that station 2 has 2 servers and station 3 has 3 servers.

By applying the proposed algorithm the three service centers can be translated into GSPN models that are eventually composed and connected according to the routing matrix, as described in the previous section. Then we obtain the overall GSPN equivalent to the given QN, as showed in Figure 4 (b). The parameters of the GSPN are completely defined by the various steps of the algorithm.

As the three blocks hold \( M \Rightarrow M \) property, we can state that the whole system has a product-form stationary probabilities function. Then the derived GSPN can be analyzed by product-form solution or by simulation. Note that in this example we have showed how it is possible to deal with class switching and no-BCMP queueing disciplines.

**CONCLUSIONS**

In this paper we have defined an algorithm that given a BCMP QN returns an equivalent GSPN. The algorithm computational complexity is linear with the number of stations of the QN and with number of no-zero elements of its routing probability matrix (in the worst case, without class switching it is of \( O(N^2 R) \) operations, and with class switching it is \( O((NR)^2) \) operations, where \( N \) is the number of the QN stations and \( R \) the number of classes). The proposed algorithm is defined in terms of a mathematical de nition of the models, it can be easily rewritten in order to deal with well-de ned languages for representing PNs and their extensions such as PNML Weber and Kindler (2003). Further research can have two directions. From the theoretical viewpoint an open problem is the de nition of general structural sufficient conditions on GSPN models that ensure that a model holds the \( M \Rightarrow M \) property. This could allow an automatic veri cation of the conditions to combine a GSPN block with others ones holding \( M \Rightarrow M \) property to obtain a product-form model. From the practical viewpoint, open research concerns possible language extensions to represent GSPNs in order to be able to represent the following features:

- to represent the concept of class of a place. Note that this does not necessarily require the idea of color, as de ned in Colored Petri Net extension.
- to identify whether a model holds \( M \Rightarrow M \) property. An open problem is the de nition of an automatic e cient algorithm to decide this condition.
- to represent the stationary state probability expression of the model in isolation for each GSPN model. In fact, although we know that a GSPN model holding the \( M \Rightarrow M \) property has a product-form solution, only if the explicit expression of the product-form is known we can obtain the stationary state probabilities for the whole net. When the product-form expression is not known, the model can be still studied by simulation, and the theoretical results guarantee that the performance indices are the same of the original hybrid model.

**REFERENCES**


Balgo G.; Bruell S.C.; and Sereno M., 2003. *On the rela-
Figure 4: (a) System modelled by a no-BCMP queueing network. (b) System modelled by a product-form GSPN.


TIMED STOCHASTIC PETRI NET MODELS IDENTIFICATION FOR SIMULATION AND RELIABILITY ANALYSIS

Souleiman Ou'd El Medhi
Edouard Leclercq
Dimitri Lefebvre
GREAH - Université le Havre
25 rue P. Lebon,
76058 Le Havre Cedex, France 3
E-mail: {dimitri.lefebvre, edouard.leclercq, souleiman.ould}@univ-lehavre.fr

KEYWORDS
Simulation, reliability analysis, Stochastic Petri net, Markov model, identification.

ABSTRACT
Reliability studies for complex dynamical systems are often based on simulation and analysis of timed stochastic Petri net models. The structure and parameters of these models are mainly unknown or partially known as long as fault processes lead to unexpected behaviours. In that case, identification methods based on the analysis of collected event sequences are of great interest. The contribution of this paper concerns the identification of timed stochastic Petri net models. Stochastic and deterministic stochastic Petri nets with deterministic and exponentially distributed random variables are considered. A systematic identification method is proposed according to event sequences that are recorded by supervision. This method is based on the idea that the dynamic behaviour of the considered PN can be mapped into a Markov model with state space isomorphic to the reachability graph of the untimed PN model.

INTRODUCTION
Reliability analysis for dynamical systems are mainly based on stochastic discrete event models as long as fault and repair processes are considered as unpredictable events. Such models are useful for simulation and analysis purposes. They are often defined as stochastic automata or Petri nets (PN). The model structure and the parameters value, are difficult to obtain when no a priori knowledge about system failures is available (Rausand and Hoyland 2004). In our preceding works (Ould El Mehdi et al. 2006), (Ould El Mehdi et al. 2007) we have investigated the synthesis of the PN structure according to the causality relationships that are derived from the occurrence order of the concerned events (Leclercq et al. 2008). In this paper, we study identification of the parameters for timed stochastic Petri net models.

Stochastic Petri net have been proved to be suitable for reliability studies, particularly for reparable systems. Among the existing extensions of stochastic PN (e.g. generalized stochastic PN, stochastic reward nets,..), stochastic Petri nets with exponentially distributed firing times (SPN) are the basic models introduced by Molloy (Molloy 1981), (Molloy 1982). Deterministic stochastic PN (DSPN) with deterministic and exponentially distributed firing times have been developed to model operations with constant duration (Ajmone and Chiola 1987). We propose a systematic method for the identification of SPN and DSPN models according to event sequences that are recorded by supervision systems (Bouché 2005), (Mannila 2002).

Two cases will be considered. The first case concerns identification of SPN models for which firing times are exponentially distributed. The identification method is based on the fact that the marking process of SPN is mapped into a Markov model with state space isomorphic to the reachability graph of the untimed PN model (Bobbio et al. 1998), (Malhis and Sanders 1996), (Nakada and Yoneyama 2000). The second case concerns identification of DSPN models with constant and exponentially distributed firing times. Identification method for DSPN is based on the design and analysis of a modified Markov model that exhibits the same steady state behaviour as the considered DSPN.

The paper has five sections. Section two introduces SPN, DSPN, definitions and assumptions. The problem is also stated. Section three is about identification of SPN. Section four is about identification of DSPN and section five is an illustrative example.

PROBLEM STATEMENT
Event sequence
An event sequence $\text{SeqE} = (\epsilon(k))_{1 \leq k \leq K}$ is defined as a row vector of size $K$. Each entry $\epsilon(k)$ represents the $k^{th}$ event of the sequence. The integer $k$ is the rank of event $\epsilon(k)$ in sequence $\text{SeqE}$. It is assumed that events $(\epsilon(k))_{1 \leq k \leq K}$ belong to $q$ distinct classes. Let $E = \{e_1, e_2, \ldots, e_q\}$ be the set of event classes and $e_j$ stands for the class $j$ in $E$. Observation of $\text{SeqE}$ associates each event $\epsilon(k)$ with its class $j$.

Let $\text{SeqT} = (t(k))_{1 \leq k \leq K}$ be a time sequence also defined as a row vector of size $K$. Each entry $t(k)$ represents the instant when the $k^{th}$ event is observed (i.e. the event $\epsilon(k)$ of rank $k$ is observed at time $t(k)$).

Stochastic and deterministic stochastic Petri nets
An ordinary Petri net (PN) is defined as $\langle P, T, W_{PR}, W_{PO} \rangle$ where $P = \{P_i\}$ is a not empty finite set of $n$ places and $T = \{T_j\}$ is a not empty finite set of $m$ transitions. Each transition $T_j$ is associated to an event class $j$. $W_{PR} = (w_{PR})_{i,j} \in \{0, 1\}^{n \times q}$ is the pre-incidence matrix and $W_{PO} = (w_{PO})_{j,i} \in \{0, 1\}^{q \times n}$ is the post-incidence matrix (David and Alla 1992). The PN incidence matrix $W$ is defined as:
\[ W = W_{P0} - W_{PR} \in \{-1, 0, 1\}^{p_x} \]  

(1)

The PN marking \( M \) is an application from the set of places \( P \) to the set of non-negative integer numbers \( Z^+ \) such that, for each place \( P_i \in P \), \( M(P_i) \) is the number of tokens in place \( P_i \). \( M_i \) is the Petri net initial marking. A firing sequence \( \sigma = T_0; T_1; \ldots; T_k \) is defined as an ordered series of transitions that are successively fired from marking \( M \) to marking \( M' \) (i.e., \( M \geq M' \)) such that equation (2) is satisfied:

\[
\sigma: M \rightarrow M_1 \rightarrow M_2 \rightarrow \ldots \rightarrow M' \tag{2}
\]

Such a sequence is represented by its characteristic vector \( X = (x_i) \in (Z^+)^q \) where \( x_i \) stands for the number of \( T_i \) firings.

The marking \( M' \) resulting from the marking \( M \) after firing the sequence \( X \) is given by \( \Delta M = M' - M = WX \).

A stochastic Petri net SPN = \( \langle PN, (\mu_1, \mu_2, \ldots, \mu_q) \rangle \) is a timed ordinary Petri net with exponentially distributed firing times and \( \mu_i \) is the firing rate of transition \( T_i \). The marking process of a SPN will be characterized according to the PN incidence matrices, the initial marking, the firing rates, the firing, server and execution policies (Bobbio et al., 1998), (Diaz 2001). SPN that are considered in this paper satisfy the following assumptions:

(H1) the firing policy is a race policy: the transition whose firing time elapses first is assumed to be the one that will fire next.

(H2) the server policy is of type infinite server: the number of firings is given by the enabling degree of the transition. The minimal period of each transition \( T_i \) is defined with a stochastic variable which is characterized according to an exponential distribution of varying parameter \( n_i(M), \mu \). The function \( n_i(M) \) is the enabling degree of transition \( T_i \) for marking \( M \).

\[
n_i(M) = \min_{P \in \Gamma_i} \{ M(P) \} \tag{3}
\]

where \( \Gamma_i \) stands for the set of \( T_i \) upstream places.

(H3) the execution policy is of type “resampling memory”: at the entrance in a marking, the remaining firing times of all transitions that are enabled is reset.

When hypothesis (H1) to (H3) are satisfied, the marking process of the SPN is mapped into a Markov model with state space isomorphic to the reachability graph of the untimed PN model (Bobbio et al., 1998), (Malhis and Sanders 1996), (Nakada and Yoneyama 2000).

A DSPN is an ordinary PN that have some transitions with constant minimal firing periods and some transitions with exponentially distributed firing periods. Let \( T_{\text{CONST}} \) be the set of transitions \( T_j \) whose minimal firing periods have constant values \( d_{\text{min}_j} \) and \( T_{\text{EXP}} \) the set of transitions \( T_j \) whose minimal firing periods are characterized according to an exponential distribution of parameter \( n_i(M), \mu_j \). Let us consider also the following assumption:

(H4) in any marking of a DSPN, at most one deterministic transition is enabled.

When hypothesis (H4) is satisfied, the marking process of the DSPN is mapped into a Markov model (Diaz 2001), (Lindemann 1993). According to (H4) there is no conflict between several transitions with constant periods. This assumption is required in order to apply the resampling memory execution policy. If conflicts occur between several transitions with constant periods, tokens will be reserved and the remaining firing times can no longer be reset in each new marking. Let us mention that assumption (H4) may be relaxed by considering non markovian marking processes (Lindemann and Thummeler 1999).

The reachability graph \( R(PN, M_i) \) of \( \langle PN, M_i \rangle \) characterises the set of states (i.e. marking \( M \)) and the set of transition sequences \( \sigma \) such that \( M \geq M' \) (the sequence \( \sigma \) is said to be executable from marking \( M_i \)). The set of states for graph \( R \) will be referred as \( S = \{ s_1, s_2, \ldots, s_p \} \) where \( p \) stands for the number of PN states, \( s_i \) stands for state \( i \) and \( s_0 \) stands for the set of transitions that can fire from state \( s_0 \).

(H5) the reachability graph of the considered PN is assumed to have a finite number of states and the Petri nets are assumed to be live.

**Problem statement**

The problem, considered in this paper is to identify SPN and DSPN parameters (i.e. firing rates and minimal firing duration) from the observation of a timed event sequence. The observed sequences SeqE and SeqT are assumed to satisfy the following assumptions:

(H6) the sequence SeqE is complete: all states \( s_i \) of the reachability graph \( R \) are visited at least once.

(H7) events are collected without errors and interruptions.

(H8) the structure of PN (i.e. incidence matrices \( W_{PR} \) and \( W_{RD} \)) is known.

If the structure of the model is not known, it is possible to identify it according to causality relationships in the sequence of events (Ould El Mehdi et al. 2006), (Ould El Mehdi et al. 2007).

**SPN IDENTIFICATION**

Let us consider in this section the identification of time parameters for SPN. The marking process is mapped into a Markov model and transitions of this process occur according to exponential distributions. The aim of this section is to identify the unknown parameters \( (\mu_1, \mu_2, \ldots, \mu_q) \) of the stochastic exponential distributions that characterise the firing periods of the transitions. These parameters will be identified from observation of sequences SeqE and SeqT. The proposed method has 3 stages:
1. To recover the sequence of successive states that are visited by the system from the sequence of observed events.
2. To estimate the mean sojourn time in states and the transition probabilities according to the Markov model and to the sequences SeqE and SeqT.
3. To identify the parameters of SPN with a mean square estimators

State sequence

The reachability graph and initial state are useful to recover the sequence SeqS of successive states that are visited by the system from the event sequence SeqE.
Let s(k) be the estimation of state during time period [t(k-1), t(k)] (i.e. following the event e(k-1) and before event e(k)). The complete and unbiased recording of event sequence SeqE leads to exact estimation s(k) of SeqS

Mean sojourn times and transition probabilities

Let us define the following sets:
- \( S_i \) = \{k such that s(k) = s_i\}: the set of ranks for state s_i in SeqS.
- \( E_j \) = \{k such that e(k) = e_j\}: the set of ranks for event e_j in SeqE.
- \( E_{i,j} \) = \{k such that e(k) = e_j and s(k) = s_i\}: the set of ranks for event e_j in SeqE when system is in state s_i.

The mean sojourn time in state s_i, estimated from the sequence SeqS, satisfies equation (4):

\[
\hat{d}_i = \frac{1}{|S_i|} \sum_{k \in S_i} (t(k) - t(k-1)) \tag{4}
\]

with \( t(0) = 0 \). When several events can occur when system is in state s_i, the probability that event e_m occurs (such probabilities will be referred in the following as “transition probabilities”) satisfies equation (5):

\[
\Pr(e_m | s_i) = \frac{|E_{i,m}|}{|S_i|} \tag{5}
\]

Identification of SPN parameters

Identification of the parameters that characterize the exponential distributions associated to the SPN transitions, results from the estimation of mean sojourn times in states and transition probabilities. The mean sojourn time in state s_i, depends on the transition firing rate \( \mu_j \) and enabling degree \( n_{i,j} = n_j(s_i) \) of transitions \( T_j \in s_i^0 \) (Rausand and Hoyland 2004):

\[
d_i = \frac{1}{\sum_{T_j \in S_i} n_{i,j}\mu_j} \tag{6}
\]

The probability that transition \( T_m \) fires when system is in state s_i is given according to equation (7):

\[
Pr(e_m | s_i) = \frac{n_{i,m}\mu_m}{\sum_{T_j \in S_i} (n_{i,j}\mu_j)} \tag{7}
\]

Estimation \( \hat{\mu} = (\hat{\mu}_1, \hat{\mu}_2, ..., \hat{\mu}_q)^T \) of \( \mu = (\mu_1, \mu_2, ..., \mu_q)^T \) is given according to equation (8):

\[
\begin{pmatrix}
A^0 \\
A^1 \\
\vdots \\
A^p \\
A \\
\hat{\mu}
\end{pmatrix} =
\begin{pmatrix}
b^0 \\
0 \\
\vdots \\
0 \\
0 \\
1
\end{pmatrix} \tag{8}
\]

with A a block matrix and b a block vector that are obtained by replacing the mean sojourn times and transition probabilities (equations (6) and (7)) with their estimations (equations (4) and (5)). The estimation \( \hat{d}_i \) of the mean sojourn time leads to p equations of the form (9):

\[
\sum_{c_i \in E_j} n_{i,j} \hat{d}_j = \frac{1}{\hat{d}_i} \tag{9}
\]

These p equations can be formulated according to a matrix equality \( A^0 \hat{\mu} = b^0 \) where \( A^i = (a_{ki}) \) is of dimension p x q with \( a_{ki} = n_{i,j} \) if \( e_j \in s_i^0 \) and 0 otherwise, and \( b^0 = (b_{ki}) \) is a column vector of dimension p whose entries are \( b_{ki} = 1 / \hat{d}_j \).
Let c be defined according to equation (10):

\[
c = \sum_{i=1}^{c_i} (|S_i| - 1) = \sum_{i=1}^{c_i} c_i \tag{10}
\]

where \( c_i \) equals the number of transitions minus one which are in conflict from the state s_i.

Estimation of transition probabilities leads to c additive equations of the form (11):

\[
(\hat{Pr}(e_m | s_i) - 1) n_{i,m} \hat{\mu}_m + \hat{Pr}(e_m | s_i) \sum_{j=0}^{p} (n_{i,j} \hat{d}_j) = 0 \tag{11}
\]

where \( e_m \) and \( e_i \) belong to \( s_i^0 \). These equations can be formulated according to a matrix equality \( A^i \hat{\mu} = 0 \) where \( A^i = (a_{ki}) \) whose dimension is \( c_i \times q \) (\( A^i \) is eventually empty) and \( a_{ki} = (\hat{Pr}(e_i | s_i) - \delta_{ki}) \) for \( k = 1, ..., c_i \), \( e_j \in s_i^0 \) stand for \( c_i - 1 \) distinct events among \( c_i \), and \( t = 1, ..., p \). Equation (8) is obtained as a consequence.

DSPN IDENTIFICATION

In this section DSPN that satisfy assumption (H4) are considered. The marking process of the DSPN is no longer a Markov process with parameters vector \( \mu = (\mu_1, \mu_2, ..., \mu_q)^T \) when conflicts exist between transitions with constant periods (TCONST) and transitions with exponentially
distributed periods \( (T_{\text{EXP}}) \). In that case, the reachability graph has some states with downstream transitions that are of both types: deterministic and also stochastic ones. In this section, a modified Markov process is introduced to characterize the marking process of the DSPN.

**Modified Markov process**

In order to introduce the modified Markov process, let us consider a state \( s \) in the reachability graph (figure 1.a) with a single deterministic downstream transitions \( T_d \) (with constant period \( d_{\text{min}} \)) and \( r \) stochastic downstream transitions \( T_{S_i} \) (with exponential distributed periods of parameters \( \mu_{S_i} \)).

All stochastic transitions \( T_{S_i} \) may fire, moreover, regarding the deterministic transition, they behave as a single stochastic transition \( T_s \) with parameter \( \mu \):

\[
\mu = \sum_{i=1}^{r} \mu_{S_i} \quad (16)
\]

So the mean sojourn time in state \( s \), as the transition probabilities, can be worked out considering the simplified situation with a single deterministic downstream transition \( T_d \) and a single stochastic downstream transition \( T_s \) (figure 1.b).

![Diagram](image)

Figure 1 : a) State \( s \) with a single deterministic downstream transition and several stochastic downstream transitions; b) state \( s \) with a single deterministic downstream transition \( T_d \) and an equivalent single stochastic downstream transition \( T_s \) of parameter \( \mu \).

The probability that \( T_s \) fires before \( T_d \) is given by equation (17):

\[
\Pr(T_s \text{ fire before } T_d) = \Pr(d(T_s) \leq d_{\text{min}}) = 1 - e^{-\mu d_{\text{min}}} \quad (17)
\]

Going back to the reachability graph in figure 1.a, one can state:

\[
\Pr(T_d | s) = e^{-\mu d_{\text{min}}} \quad \text{for } T_d \in \text{T}_{\text{CONST}}
\]

\[
\Pr(T_{S_k} | s) = \left( \frac{\mu}{\mu_{S_k}} \right) \left( 1 - e^{-\mu d_{\text{min}}} \right) \quad \text{for all } T_{S_k} \in \text{T}_{\text{EXP}}
\]

The distribution of sojourn times in state \( s \) for figure 1.b is no more exponentially distributed. The density function for sojourn time is given according to equation (19):

\[
f(t) = \mu e^{-\mu t} \quad \text{if } 0 \leq t < d_{\text{min}}
\]

\[
f(t) = \frac{e^{-\mu t}}{\Delta t} \quad \text{if } d_{\text{min}} \leq t < d_{\text{min}} + \Delta t
\]

\[
f(t) = 0 \quad \text{anywhere else}
\]

So, the mean sojourn time in state \( s \) is given according to equation (20):

\[
\hat{d} = \lim_{\Delta t \to 0} \left( \int_0^{d_{\text{min}} + \Delta t} f(t) \, dt \right) = \frac{1}{\mu} \left( 1 - e^{-\mu d_{\text{min}}} \right)
\]

In order to estimate the parameters \( d_{\text{min}} \) and \( \mu \) and thereafter parameters of stochastic transitions \( \mu_{S_1} \) to \( \mu_{S_r} \), the identification method proposed in section three is still acceptable with a modified Markov process with parameters \( \mu'_{d} \) and \( \mu'_{s} \) instead of \( d_{\text{min}} \) and \( \mu \) that exhibits the same mean behaviour (same mean sojourn time in states and the same transition probabilities). For the example described in figure 1, parameters \( \mu'_{d} \) and \( \mu'_{s} \) satisfy:

\[
\frac{\mu'_{d}}{\mu'_{d} + \mu'_{s}} = e^{-\mu' d_{\text{min}}}
\]

\[
\frac{1}{\mu'_{d} + \mu'_{s}} = \frac{1}{\mu} \left( 1 - e^{-\mu' d_{\text{min}}} \right)
\]

that leads to:

\[
\mu'_{d} = \frac{\mu e^{-\mu' d_{\text{min}}}}{\left( 1 - e^{-\mu' d_{\text{min}}} \right)}
\]

\[
\mu'_{s} = \mu
\]

**Identification of DSPN parameters**

It is easy to generalize the previous result for any DSPN when the sets \( T_{\text{CONST}} \) and \( T_{\text{EXP}} \) are a priori known and assumption (H4) is satisfied. Identification of DSPN parameters is obtained according to a three-stages procedure:

1. The marking process of DSPN is mapped into a modified Markov process with parameters vector \( \mu' = (\mu'_{1}, \mu'_{2}, \ldots, \mu'_{q'}) \) estimated by vector \( \hat{\mu}' = (\hat{\mu}'_{1}, \hat{\mu}'_{2}, \ldots, \hat{\mu}'_{q'}) \), with \( q' \geq q \). The idea is to replace the deterministic transitions by stochastic ones in order to preserve the mean behaviours. When a deterministic transition occurs in several conflicts in the reachability graph, it is replaced by several stochastic transitions with distinct parameters (a new one per conflict). For this reason \( q' \geq q \).

2. Parameters of stochastic transitions \( T_{\text{EXP}} \) are worked out with equation (23):
\[ \hat{\mu}_j = \hat{\mu}'_j \text{, for all } T_j \in T_{\text{EXP}} \] (23)

**Figure 2:** Estimation of parameters for transitions with exponential firing periods

3. Minimal firing periods of constant transitions \( T_{\text{CONST}} \) are worked out with equations (24) to (28) according to the following cases:

3.a. If transition \( T_h \in T_{\text{CONST}} \) is the single downstream transition of \( m_{ns} \) states \( s_k \) \((m_{ns} \text{ conflicts between } T_h \text{ and one or several stochastic transitions})\) of the reachability graph, then the parameter \( \hat{d}_{\text{min}} \) will be estimated according to a single parameter \( \hat{\mu}'_h \):

\[ \hat{d}_{\text{min}} = 1/\hat{\mu}'_h \] (24)

**Figure 3:** Estimation of parameters for transitions, with constant firing periods, that are the single downstream transitions of one or several states

3.b If transition \( T_h \in T_{\text{CONST}} \) is the non single downstream transition of \( m_{ns} \) states \( s_k \) \((m_{ns} \text{ conflicts between } T_h \text{ and one or several stochastic transitions})\) of the reachability graph, then the parameter \( \hat{d}_{\text{min}} \) will be estimated according to a weighted sum depending on the identification of \( m_{ns} \) additional parameters \( \hat{\mu}'_{1,h} \) to \( \hat{\mu}'_{m_{ns},h} \):

\[ \hat{d}_{\text{min}} = \left( \sum_{s_k=h} \alpha_{k,h} \ln \left( \frac{\sum_{T_{\text{sub}}} n_{s_k} \hat{\mu}_h + \hat{\mu}'_{k,h}}{\hat{\mu}'_h} \right) \right) + \alpha_h \] (27)

**Figure 5:** Estimation of parameters for transitions, with constant firing periods, that are the single downstream transitions of one or several states and also appear in several conflicts

where \( \alpha_{k,h} \) and \( \alpha_h \) stand, respectively, for the ratio of \( T_h \) firings following state \( s_k \) with several downstream transitions and for the ratio of \( T_h \) firings following states with a single downstream transition:

\[ \alpha_{k,h} = \frac{|E_{k,h}|}{|E_h|}, \text{ for } s_k \in \sigma T_h \text{ and } |s_k| > 1 \]

\[ \alpha_h = \sum_{s_k=h \text{ and } |s_k|=1} \frac{|E_{k,h}|}{|E_h|} \] (28)

To apply the proposed method, one must separate at first the stochastic transitions and the constant ones. If this classification is a priori not possible, statistical tests as Chi-square-test can be used for classification purposes.

**EXAMPLE**

Let consider the PN in figure 6, with 10 places and 12 transitions. This PN is a simple model that represents a manufacturing system with two sequential lines \( T_1 - P_1 - T_2 \)
other are stochastic ones (DSPN). Four transition firings (T₁, T₂, T₉, T₁₀, T₁₁) occur according to deterministic periods. The minimal periods of all other transitions are defined according to exponentially distributed variables (table 1).

The results that are discussed in this section are obtained according to a simulation sequence S’ is of size 100 000 TU. The estimated values of the parameters must be identified according to the sequence S’. In order to apply the algorithm in section 4, a modified Markov process with 6 additional parameters is introduced. These parameters are added in order to take into account the influence of deterministic transitions appearing in conflicts. The transition T₃ is the single output transition of a single state and does not appear anywhere else in the reachability graph. The transition T₇ appears in a single conflict after a single state. Thus, periods \( \hat{d}_{\text{min}_3} \) of transitions T₃ and \( \hat{d}_{\text{min}_7} \) of transition T₇ will be estimated according to new parameters \( \hat{\mu}_1 \) and \( \hat{\mu}_7 \). The transition T₂ is the single output transition of one state and appears also in 3 conflicts following 3 distinct states. Thus, parameter \( \hat{d}_{\text{min}_2} \) of transition T₂ will be estimated according to 4 new parameters. The same holds for transition T₉. Thus, parameter \( \hat{d}_{\text{min}_9} \) of transition T₉ will be estimated according to 4 new other parameters. Analysis of sequence S’ and estimation of stochastic and deterministic transitions parameters result from the two-stages estimation algorithm described in section 4. Results are provided in table 1:

![Diagram of a manufacturing system](image)

**Figure 6: PN model of a manufacturing system**

The reachability graph of the PN in figure 6 has 24 states and 12 events. Identification of parameters will be considered when some transitions are deterministic ones and

<table>
<thead>
<tr>
<th>Table 1 : Estimation of DSPN parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected value</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Estimated value</td>
</tr>
<tr>
<td>Error</td>
</tr>
</tbody>
</table>

![Graph](image)

**Figure 7. Estimation of DSPN parameters with respect to time for the period 1 000 TU – 18 000 TU**

(a) evolution of the parameter \( \hat{d}_{\text{min}_2} \) (b) evolution of the parameter \( \hat{\mu}_9 \)
Figure 7 illustrates evolution of the estimated value of parameters for transitions $T_5$ and $T_9$ with respect to time for the period 1 000 TU – 18 000 TU. Figure 8 illustrates the evolution of the structure of matrix $A$ with respect to time during the same period.

Estimation results are quite good, excepting the estimation of parameters for rare events ($e_{11}$) that is poor and must be improved (Table 1). The estimations of parameters depend on the number of firings of each transition from each state this transition is in conflict. In addition to the sequence size, the relative frequency of the most complex transition (a transition in conflict from different states) will influence the performance.

![Structure of matrix A](image)

![Minimal number of firings for transitions](image)

Figure 8. Evolution of the structure of matrix $A$ with respect to time for the period 1 000 TU – 18 000 TU (a) Number of visited states (full line); number of transition probability equations (dashed line); rank of matrix $A$ (strong dotted line) (b) Minimal number of firings for the transitions

The figure 9 depicts the quadratic estimation error for the parameter of transition $T_2$ with respect to the minimal firings number over all transitions ($T_4$, $T_8$, $T_9$, $T_{10}$) in conflict with $T_2$. Not only the structure of the matrix $A$ and the number of firings of the transitions are important factors that explain the performance, but one must also take care of all transitions that appear in the same conflict. In other words, the parameter of a transition that fires many time but that is in conflict with another transition that corresponds to a rare event and fires only a few times will be difficult to estimate with a good accuracy.

![Quadratic error for estimation of parameter of $T_2$](image)

Figure 9. Quadratic estimation error for the parameter of transition $T_2$ with respect to the minimal firings number over all transitions in conflict with $T_2$.

The estimation of the parameter related to the rare event ($e_9$) is better than the estimation of the parameter related to the rare event ($e_{11}$) because more states are visited with $T_9$ as a downstream transition, than with transition $T_{11}$.

**CONCLUSIONS**

The main contribution of this paper is to propose a method for the identification of stochastic and deterministic stochastic Petri net models. Time parameters of the transitions are estimated according to the observation of event sequences that are recorded by supervision systems. The contribution is of particular interest for the reliability analysis of repairable systems. The reason is that failure rates are often unknown and the method leads to an estimation of these parameters. Numerous perspectives will be studied in our future works.

Occurrences of errors or interruptions in the observation sequences will also be considered. When only the event sequence is observed, directed paths in the PN graph (Lefebvre and Delheur 2007) can be used to recover the wrong or missing events. When the event and state sequences are both observed, estimation methods based for example on coding theory (Lefebvre 2008) will be combined with the identification algorithm.

Due to the combinatorial explosion of the number of states, the proposed method is not suitable for large systems (the dimension of matrix $A$ increases as the size of the underlying Markov model). In that case, decomposition or fluidification (partial or complete) method will be applied to the Petri net models in order to simplify the identification method (Silva and Recalde 2004), (Trivedi and Kulkarni 1993).

Identification may be improved with a multi-stages algorithm. The idea is to use at first only states which are often visited and conflicts that fire at most in order to obtain a partial identification of the parameters for frequent events. Then, remaining parameters concerning rare events could be estimated with other states and conflicts and according to the estimated value of frequent events. Our attention will focus
particularly on the identification of parameters that characterize rare events. Statistical tests will also be investigated for that purpose.

At last, identification method will also be extended to other stochastic distributions (for example, normal distributions) in order to deal with a large variety of problems. Identification of DSPN with enabling and aging memory execution policies and of generalized stochastic Petri nets with immediate transitions will also be considered in our further works.

Application of this technique to a real-life process will allow us to improve and to validate our method.

REFERENCES


Mannila H. "Local and global methods in data mining: basic techniques and open problems". In P. W. et al., editor, 29th International Colloquium on Automata, Languages, and Programming, volume 2380 of Lecture Notes in Computer Science, pp. 57-68. Springer-Verlag, 2002.

Molloy M.K., On the integration of delay and throughput in distributed processing models, Ph. D, UCLA, Los Angeles, USA, September 1981


PROJECT MANAGEMENT
Object Oriented Petri Nets Usage in Project Portfolio Modeling Domain

Sarka Kvetonova
Vladimir Janousek
Faculty of Information Technology
Brno University of Technology
Bozetechova 2
612 66 Brno, Czech Republic
E-mail: {kvetona, janousek}@fit.vutbr.cz

KEYWORDS
Project, Project Portfolio Management, Object Oriented Petri Nets, Optimization.

ABSTRACT
The paper presents a way, how the Petri net technique, in concrete, Object Oriented Petri Nets (OO PN) can be applied in project management domain. There is shown the use of Object Oriented approach and concepts in project planning, organizing, monitoring and control phases of a project. Then, there is described, in more details, which important properties or artefacts of OO PN are useful for control and online monitoring and optimization of project portfolio.

INTRODUCTION
The need of efficient project control is required together with increasing complexity of project resource consumption (by project, we mean a collection of mutually bound activities that build up a process). The most used and popular graphic representation or technique used for project planning by these days is based on different types of diagrams; we can say that those are methods of net analysis used as a tool for project analysis and control. Model of a project, in such a case, is the network diagram (simple directed acyclic graph with a single start node and a single end node).

To date there exist number of network analysis methods, firstly, CPM (Critical Path Methods), PERT (Program Evaluation and Review Technique), GERT (Graphical Evaluation and Review Technique), MPM (Metra Potential Method), and others. These techniques are successful in an “offline” model of planning and monitoring - it is quite difficult to monitor and control project (plan) performance and, moreover, to model and detect resource limitations. One of the possibilities to gap this limitation is the use of Petri net technique, in concrete, Timed Coloured Petri nets, see (Květoňová, Kolář 2008). There is proposed an algorithm, how the key main network diagram representations of a project plan were added next.

If we targeted on relation mentioned above, we can use Petri nets techniques by different, possibly to say, more suitable and/or more hands-on purpose enable us to online control and optimize the whole project portfolio. Event if important aspects of some project are changed at this time.

We use the object oriented approach Petri nets model and online monitoring project portfolio. We resume the main disadvantages/advantages of actually used approaches and we propose some new and more sophisticated possibilities of it.

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 the following section the main foundations will be given that are crucial for expression and comprehension of a mutual context among terms of Project management, planning/scheduling and Petri Nets.

TECHNICAL BACKGROUND
First of all, some terms from management and controlling are presented. Nevertheless, it is expected that a reader is familiar with the concept. Deeper explanation of the terms can be found in (Anbari 2005, Gray and Larson 2005, Kerzner 2005, Morris and Pinto 2007, Spinner 1992, Girault 2003, Rudová 2006 etc.).

Project management

A project is a temporary effort undertaken to create a unique product or service, or result conforming to certain specifications and applicable standards. See (P.M.I. 2004, Anbari 2005).

Project management is a procedure of managing and directing time, material, personnel and costs to complete a particular project in an orderly and economical manner; and to meet established objectives in time, costs, and technical results. See (Spinner 1992). 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. See (P.M.I. [online]; 2004, Schwalbe 2005).

A 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. At present, activities and resources under the project are managed almost entirely like processes. See (Anbari 2005).

Project Portfolio Management is about more than running multiple projects. Each portfolio of projects needs to be assessed in terms of its business value and adherence to strategy. The portfolio should be designed to achieve a defined business objective or benefit. Project Management guru Bob Buttrick put it well when he said: "Directing the individual project correctly will ensure it is done right. Directing 'all the projects' successfully will ensure we are doing the right projects."

Planning and scheduling

Planning and scheduling are processes dealing with tasks or activities scheduling in time and space. Their main goal is to gain necessary tasks scheduling on limited resources by use of mathematic techniques and heuristic methods. Planning is a long-run process of a set of proper activities creation to gain the predefined goals. Scheduling is the process of converting a general or outline plan for a project into a time-based graphic presentation given information on available resources and time constraints.

A task is an object which is planned. It is characterized by own properties and inherent structure. An operation or subtask, too, is a partial part of the whole task. The task consists of one or more operations which can be realized on one or more resources.

Resource is processing and/or it realizes the individual operations, eventually, it serves as means an operation realization. Resources use to be limited in a system.

Static planning and scheduling means, we know all aspects of a schedule, all exercises into scheduling and all other restrictions or demands on it. Any other information does not come into a system. We know all accessible resources, too. Thus, the whole schedule is created or planned at a time at the same instant of time. In other words, static planning represents so called off-line planning (Off-line planning describes a process of a schedule production before a system starting, thus, before tasks start work on resources). Reactive planning denotes a group of techniques for selection of an action by autonomous agents. These techniques differ from classical planning in two aspects. First, they operate in a timely fashion and hence can cope with highly dynamic and unpredictable environments. Second, they compute just one next action in every instant, based on the current context. Reactive planners often (but not always) exploit reactive plans, which are stored structures describing the agent's priorities and behaviour. The term reactive has now become a pejorative used as an antonym for proactive. Since nearly all agents using reactive planning are proactive, some researchers have begun referring to reactive planning as dynamic planning.

A timetable/schedule is an organized list, usually set out in tabular form, providing information about a series of arranged events: in particular, the time at which it is planned these events will take place.

Petri Nets

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).

Definition 1: A net is a triple \( N = (P, T, F) \) where P and T are mutually disjoint sets such that:
- \( P = \{p_1, p_2, ..., p_n\} \) is a finite set of places
- \( T = \{t_1, t_2, ..., t_m\} \) is a finite set of transitions
- \( F \) is binary flow relation: \( F \subseteq (P \times T) \cup (T \times P) \)

Definition 2: P/T Petri net, N, is a six-tuple \( N = (P, T, F, W, K, M_0) \), where:
- \( (P, T, F) \) is a finite net
- \( W: F \rightarrow (\mathbb{N} \setminus \{0\}) \) is a non-negative weight of every net edge.
- \( K: P \rightarrow (\mathbb{N} \cup \{\omega\}) \) is a map denoting capacity of a place.
- \( M_0: P \rightarrow (\mathbb{N} \cup \{\omega\}) \) is initial marking of the Petri Net such that \( \forall p \in P: M_0(p) \leq K(p) \)
- \( N \) is a set defined as \( N = \{0,1,2,...\} \)
- \( \omega \) is a supremum of the net \( N \) with the following features:
  - \( n \in N: n < \omega \)
  - \( m \in (N \cup \{\omega\}): m + \omega = \omega + m = \omega - m = \omega \)

Next, if \((p, t) \in F \cap (P \times T)\), then we say that \( p \) is the entry place and \( (p, t) \) is the entry edge of the transition \( t \). Similarly, if \((t, p) \in F \cap (T \times P)\), then we say that \( p \) is the exit place and \((t, p)\) is the exit edge of the transition \( t \).

Definition 3: Let \( N = (P, T, F) \) is a net. For all \( x \in (P \cup T) \) we define:
- \( x_{\text{pre}} = \{y | Fxy\} \) to be a preset of the element \( x \).
- \( x_{\text{post}} = \{y | Fyx\} \) to be a postset of the element \( x \).
**Definition 4:** Let \( N = (P, T, F, W, K, M_0) \) be a P/T Petri net then

- mapping \( M: P (N \cup \{o_i\}) \) is called marking of the Petri Net \( N \) if \( \forall p \in P: M(p) \leq K(p) \)
- Let \( M \) be a marking of a Petri Net \( N \). Transition \( t \in T \), \( t \in T \), can fire in marking \( M \) if:
  - \( p \in t_{\text{pre}} : M(p) \geq W(p, t) \)
  - \( p \in t_{\text{post}} : M(p) \leq K(p) - W(t, p) \)

**Definition 5:** Let \( N = (P, T, F, W, K, M_0) \) be a P/T Petri net and \( M \) is a marking. If \( t \in T \) is enabled in marking \( M \) then it can be executed. By its execution we get new marking \( M' \) which is defined as follows: \( \forall p \in P: M'(p) = \)

- \( M(p) - W(p, t) \) if \( p \in t_{\text{pre}} \cap \text{tpost} \)
- \( M(p) + W(t, p) \) if \( p \in t_{\text{post}} \setminus t_{\text{pre}} \)
- \( M(p) - W(p, t) + W(t, p) \) if \( p \in t_{\text{pre}} \cap t_{\text{post}} \)
- \( M(p) \) if other.

T-transition firing from marking \( M \) into marking \( M' \) we denote by symbolic way \( M \rightarrow M' \).

A status of a system modelled by a Petri Net is denoted by a marking of such a Petri Net. A dynamic behaviour of a modelled system is denoted by the transition firing. Influence of a Petri net marking on the possibility of transition firing and, vice versa, influence of transition firing on the Petri net marking evolution is obvious. 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. 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.

In the next part of the submission we focus on Object Oriented Petri Nets concept that is crucial for our approach. See (Janoušek, 1995).

**OBJECT ORIENTED PETRI NETS**

Object Oriented Petri Nets (OOPN) consist of Petri nets organised in classes. Each class consists of an object net and a set of dynamically instantiable method nets. Places of the object net are accessible for transitions of method nets. Object nets as well as method nets can be inherited. Inherited transitions and places of object nets (identified by their names) can be redefined and new places and/or transitions can be added in subclasses. Inherited methods can be redefined and new methods can be added in subclasses. Classes can also define special methods called synchronous ports, which allow for synchronous interactions of objects. Message sending and object creations are specified as actions attached to transitions. Execution of transitions is polymorphic --- methods which will be invoked are chosen according to the classes of message receivers that are known at the compile time. A token in OOPN represents either a trivial object (e.g., a number or a string) or an instance of some Petri net-described class consisting of one instance of the appropriate object net and possibly several concurrently running instances of invoked method nets.

![Fig.1: The OOPN example](image)

Object-orientation of PNTalk and the associated OOPN formalism is based on the well-known, class-based approach in Smalltalk-like style (Goldberg 1989). It means that all objects are instances of classes, every computation is realized by message sending, and variables can contain references to objects. A class defines behaviour of its instances as a set of methods (they specify reactions to received messages). A class is defined incrementally, as a subclass of some existing class. In OOPN, this classical kind of object-orientation is enriched by concurrency. Concurrency of OOPN is accomplished by viewing objects as active servers. They offer reentrant services to other objects and at the same time they can perform their own independent activities. Services provided by the objects as well as the independent activities of the objects are described by means of high-level Petri nets - services by method nets, object activities by object nets. Tokens in nets are references to objects. Apart from the concurrency of particular nets, the finest grains of concurrency in OOPN are the transitions themselves (they represent concurrency inside a method or object net).

An example illustrating the OOPN formalism is shown in Figure 1. As it is depicted in Figure 1, a place can be inscribed by an initial marking (a multiset of objects) and an initial action (allowing a creation and initialization of complex objects to be initially stored in the place; not shown in the Figure 1). A transition can have a guard restricting its fireability and an action to be performed whenever the transition is fired. Finally, arcs are inscribed by multiset expressions specifying multisets of tokens to be moved from/to certain places by the arcs associated with a transition being fired.

The OOPN on the Figure 1 demonstrates that the method nets of a given class can share access to the appropriate object net - the places of the object net are accessible from transitions belonging to a method nets. In this way the execution of methods can modify the state of the object. The class Main describes an active object, which can instantiate (and communicate with) a passive object - stack (an object is passive if its object net contains no transitions). Each method net has parameter places and a return place. These places are used for passing data (references to objects) between calling transition and the method net. Apart from method nets, classes can also define special methods called synchronous ports that allow for synchronous interactions of objects. This form of communication (together with execution of the appropriate transition and synchronous port) is possible when the
calling transition (which calls a synchronous port from its
guard) and the called synchronous port are executable
simultaneously.

The transition guards and actions can send messages to
objects. An object can be either primitive (such as number
or string), or non-primitive (defined by Petri nets). The way
how transitions are executed depends on transition actions.
A message that is sent to a primitive object is evaluated
atomically (thus the transition is also executed as a single
event), contrary to a message that is sent to a non-primitive
object. In the latter case, the input part of the transition is
performed and, at the same time, the transition sends the
message. Then it waits for the result. When the result is
available, the output part of the transition can be performed.
In the case of the transition guard, the message sending has
to be evaluable atomically. Thus, the message sending is
restricted only to primitive objects, or to non-primitive
objects with appropriate synchronous ports.

OUR APPROACH

Actually, there exist many approaches, methods and
formalisms to project portfolio management, e.g. see
(Kendall 2003, Archer 2004, Yan 2003). But, an interesting
possibility which is still neglected nowadays is object
oriented principles together with OO PN use to both
planning, monitoring etc. and scheduling of a project
portfolio. In a both examples is necessary to implement it
by a suitable way to realize not only project planning,
organizing etc., but optimization or dynamic modification
of project parameters depending on the actual variable
external conditions (changes in resources structure, in
project plans etc.).

The main advantages of OOPN we see in a more precise
modeling and well controlled process of project portfolio
management. We can plan, organize, control and optimize,
too, some important parameters of individual projects, if
necessary.

There are several approaches which combine Petri nets and
objects. Especially Reference nets (Valk 1998) are very
promising because of their theoretical background allowing
for analysis and verification. Nevertheless, we will
concentrate to Object Oriented Petri nets (Janousek 1995),
which are very interesting for the project portfolio
modelling domain because the formalism offers the concept
of dynamically instantiable method nets and shared places
belonging to an object net. This feature allows for
straightforward modelling of resources shared among a set
of running projects (processes).

Our approach is based on the following ideas:

- Whole project portfolio is modeled by the only OOPN
  object. Tokens in places of an object net model the
  shared resources – they are distributed in places
  according to their roles.

- Method nets correspond to individual project plan
  templates. Their instances can dynamically come up or
die down (it corresponds to project start and/or ending)
and they share an access to the object net's places
containing resource objects. Start of a project is
modeled by an appropriate message sent to the object,
possibly with parameters. At the same time a new
instance of the method net (i.e. project plan) is created
and starts to run.

An example of Project Portfolio Modelling

An example of the approach is shown on Fig. 2. There are
described three different project templates (OOPN
methods) which share the same resources. Actually, the
resource types (their roles) are modelled by places of an
object net). Project templates A and B are collapsed (their
structure is not shown). The transitions containing
expression of type self projectA: x are used for invocation
of the corresponding methods, i.e. for creating individual
projects. Note, that is possible to invoke a method several
times and they can overlap in time.

Actually, there are not all the details shown in the figure.
Especially, the individual resources are not considered yet.
Resources are modelled by individual objects which are
available as tokens in the places. Actually, the tokens are
references to the objects. So it is possible to have a resource
available under two and more roles modelled by places.
Figure 3 depicts a situation where an individual object
defined by class Resource is shared by two projects.

Each project needs a resource of particular type/role. In
both the places are tokens pointing to the same resource
object. Each activity is modelled by two transitions, one of
them seizure the resource and the second one releases it. It is
specified by the corresponding synchronous ports calling.
Besides the resource seizing and releasing the guards of the activity transitions can also specify time constraints and scheduling constraints (by querying resources and their schedule).

**Practical application for scheduling and monitoring**

For planning and monitoring we suppose simulation synchronized by real time and by events from real surrounding. When a project of a pre-defined type starts corresponding method is invoked. From a technical view, this is realized by sending of a message with defined parameters.

Within monitoring, resources (modelled as tokens in places of the object net) have assigned the concrete schedule. Transitions guards modelling individual activities have to check (besides the immediate physical availability) the fact, whether a schedule allows to assign the resource to an activity.

If a structure of resources changes it is necessary, immediately, to edit places and tokens which model this resources (automatically). Then is realized the new planning and scheduling. If a schedule is created the objects of resources are updated (they “remember” schedules of individual activities). For scheduling is created a clone of an actual state. This is used as an initial state for searching of possible futures with various ways of resource sharing/allocation based on selected optimization strategy and chosen criteria.

Actual state is available as structured – state of resources is modeled by places and tokens in its, state of a project is modeled by marking of a corresponding instance of corresponding project type (method net). It is possible to use it to visualization of the state or for next analysis.

Both a net modeling a project and individual instances can be modified in course of analysis accordance with variable external requirements. Creation of a new project type means an addition of a new method in run time.

This way of OOPN use supposes we need to extend OOPN model with reflective interface (metamodel protocol, MOP) which enables us to become involved in OOPN structure in run. This enhancement has already been proposed and experimentally implemented in a language and system Pntalk that is an OOPN implementation in Smalltalk (Janoušek 2003). OOPN/Pntalk MOP allows for inspecting and edition of particular nets defining classes individual net instances implementing the actually living objects and actually running method invocations.

It also allows to make a clone of the whole running system and to store/restore it to/from database. Without these features it was hardly possible to use it in practice.

OOPN is possible to implement in some other way, e.g. as the database application. As an alternative way we plan to use the CPN Tools where is possible to use its possibilities of analysis and verification. In this case, there is necessary to transform OOPN into CPN. For this purpose, thus, for Petri net transfer between Pntalk (and/or other OOPN variant) and CPNTools will be used PNML language (The Petri Net Markup Language (PNML) [online]).

**CONCLUSION**

The paper resumed actual possibilities of project planning, monitoring, control and optimization by way of special algorithms, methods and techniques application. The submission presented how we can use Petri nets technique to online monitoring and control of project portfolio that means optimization of a large number of different projects with the same human resources or with other types of it.

We demonstrated the main advantages of OO Petri nets in project portfolio modelling domain and we tried to foreshadow possibilities of the next/further research in this domain. The main purpose is a simplification the whole process of project portfolio control, in concrete, the online monitoring system implementation that enable us to modify important parameters if necessary or if require it the actual situation.

We can summarize the ways how to model project portfolio by Petri nets as follows:

- Obvious solution is to maintain one global High-level Petri net model of the whole portfolio.
- OOPN-based models can use dynamically instantiable nets for project modeling – static pre-defined projects planes are possible to instantiate dynamically, instances can be parametrized. However, projects plans are statically fixed.
- Suggested solution: Instantiable project petri nets with a possibility to edit their structure at runtime. Thus, the project plans can be modified arbitrarily while the structured form of the model is still preserved. We work with the pre-defined types of project plans as well as with individual instances. Thus, it has to be possible to edit both project types, and individual instances.

In the future, we suppose that the proposed model will be used for project planning, scheduling and optimization support and will facilitate the whole process of the project portfolio management by controlling (online monitoring) all important sub-processes and their parts.

**ACKNOWLEDGEMENTS**

This research was supported by the Research Plan No. MSM 0021630528, Security-Oriented Research in Information Technology.

**REFERENCES**


Goldberg A. Smalltalk 80: The Language. Addison-Wesley, 1989.


PNML - The Petri Net Markup Language [online] http://www2.informatik.hu-berlin.de

Planning and scheduling [online] http://www.maxwideman.com

Project Management Institute - P.M.I. [online] www.pmi.org

Project Portfolio Management [online] http://www.projectsmap.co.uk


AUTHOR BIOGRAPHY

ŠÁRKA KVĚTOŇOVÁ 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.

VLADMÍR JANOUŠEK was born in 1966, Boskovice, Czech Republic. He received the Ph.D. degree from the Faculty of Information Technology, Brno University of Technology in 1999. He is an assistant professor in the Department of Intelligent Systems at the Faculty of Information Technology, Brno University of Technology. His research focuses on simulation-driven development, pure object orientation and reflective architectures.
TIMED COLOURED PETRI NETS USE IN PROJECT MANAGEMENT FORMALIZATION DOMAIN

Šárka Květoňová
Dušan Kolář
Faculty of Information Technology
Brno University of Technology
Božetěckova 2
612 66 Brno, Czech Republic
E-mail: {kvetona, kolar}@fit.vutbr.cz

KEYWORDS
Project, Project Management, Project Plan, Timed Coloured Petri Nets, Formalization.

ABSTRACT
The aim of this paper is to present a unified approach to Petri Nets exploitation for project plan creation and modelling needs. It describes how a technique of Petri Nets, in concrete, the Timed Coloured Petri Nets (T-CPN) can be used to model and, subsequently, to online monitor project plan development. Our intention is to simplify the whole process of project plan creation and control through the Petri nets models by way of online optimization of a project in a discrete time. The created T-CPN model is extended, in addition, with resource modelling and mutually exclusive activity performance modelling. We have also introduced a new algorithm, which is able to translate and, as a next step, to simplify main network diagram representations to a T-CPN. If required, the resource modelling can be added.

INTRODUCTION
It is quite well accepted that the need of efficient project control is required together with increasing complexity of project resource consumption (for project, see [Anbari 2005, P.M.I. 2004]). The most used and popular technique used for project planning by these days is based on a network diagram; we can say that those are methods of net analysis used as a tool for project analysis and control. Model of a project, in such a case, is the network diagram (see [Veber 2006]).

There are several methods of network analysis already: CPM (Critical Path Methods), PERT, GERT, MPM, and others. These techniques are successful in an “offline” model of planning and monitoring - it is quite difficult to monitor and control project (plan) performance and, moreover, to model and detect resource limitations. Thus, it is important to find other ways and possibilities how to monitor and control whole project realization. Especially, if any unexpected events occur is necessary to modify some of project parameters immediately (through the online support of project management).

We have introduced an algorithm for transformation of edge-oriented network diagram to a Timed Coloured Petri Nets (T-CPN). See (Květoňová, Kolář 2008).

We were using such a created model for a certain period and started to tweak the project features. We discovered that some parts of the model are unnecessarily complex - this is due to algorithm that creates model in a universal way, so that it could be used under all circumstances. Thus, we started to search for an optimization.

The paper presents a formal definition of shadow places present in the T-CPN. Such places, if added to a network, do not modify behaviour of the T-CPN - a proof correctness is presented. Thus, if such a place is identified in the network (which is quite easy) it can be eliminated and the T-CPN simplified.

The paper starts with preliminary definitions necessary for straightforward reading and understanding of the notions used then it presents main contribution and is closed with conclusions.

TECHNICAL BACKGROUND

Processes, Project management

First of all, some terms from management and controlling are presented. Nevertheless, it is expected that a reader is familiar with the concept. Deeper explanation of the terms can be found in (Anbari 2005, Gray 2005, Kerzner 2005, Morris 2007, Spinner 1992).

Definition 1. A project is a temporary effort undertaken to create a unique product or service, or result conforming to certain specifications and applicable standards. See (Anbari 2005, P.M.I. 2004).

Definition 2. A 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. At present, activities and resources under the project are manager almost entirely like processes. See (Anbari 2005).

A graphical view on a project is usually provided through network diagram.

Definition 3. A project network diagram, shortly network diagram, is a graphical representation of actions sequences that may compose a project performance. A network diagram shows flow of actions in time during process development. The diagram enables ordering of actions in a sequence or in parallel branches, it shows their mutual collisions, consequences, inefficiencies, time gaps, overall process length, etc. See (Veber 2006).
Definition 4. A network, $S$, is a quadruple $S = (G, z, s, c)$, where $G$ is an oriented graph, $z, s \in V(G)$ are two special vertices of graph $G$ - there is no input edge of the vertex $z$ and no output edge out of the vertex $s$, $c$ is a function, $c : E(G) \rightarrow N$, that assigns to every oriented edge $e$, $e \in E(G)$, a natural number $c(e)$.

The vertex $z$ is called source (or start), the vertex $s$ is called sink (or end), and the number $c(e)$ is called capacity (or throughput) of the edge $e$.

Petri Nets

The following definitions introduce necessary background to Petri Net theory. Broader and deeper explanation can be found in Češka et al., Hirault 2003, Rozenberg 1991.

Definition 5: A net is a triple $N = (P, T, F)$ such that $P$ and $T$ are mutually disjoint sets such that:

- $P = \{p_1, p_2, ..., p_n\}$ is a finite set of places
- $T = \{t_1, t_2, ..., t_m\}$ is a finite set of transitions
- $F$ is a binary relation: $F \subseteq (P \times T) \cup (T \times P)$

Definition 6: $P/T$ Petri net, $N$, is a six-tuple $N = (P, T, F, W, K, M_0)$, where:

- $(P, T, F)$ is a finite net
- $W : F \rightarrow (N \setminus \{0\})$ is a non-negative weight of every net edge
- $K : P \rightarrow (N \cup \{0\})$ is a mapping denoting capacity of a place.
- $M_0 : P \rightarrow (N \cup \{0\})$ is initial marking of the Petri net such that $\forall p \in P: M_0(p) \leq K(p)$
- $N$ is a set defined as $N = \{0, 1, 2, ...\}$
- $\omega$ is a supremum of the net $N$ with the following features:
  - $n \in N$, $n < \omega$
  - $m \in (N \cup \{0\})$: $m + \omega = \omega + m = \omega - m = m$.

Next, if $(p, t) \in F \cap (P \times T)$, then we say that $p$ is the entry place and $(p, t)$ is the entry edge of the transition $t$. Similarly, if $(t, p) \in F \cap (P \times T)$, then we say that $p$ is the exit place and $(p, t)$ is the exit edge of the transition $t$.

Definition 7: Let $N = (P, T, F)$ be a net. For all $x \in (P \cup T)$ we define:

$\text{x_{pre}} = \{y | yFx\}$ to be a preset of the element $x$,

$\text{x_{post}} = \{y | xFy\}$ to be a postset of the element $x$.

Definition 8: Let $N = (P, T, F, W, K, M_0)$ be a $P/T$ Petri net

- $\text{M} : P \rightarrow (N \cup \{0\})$ is called marking of the Petri Net $N$ if $\forall p \in P : M(p) \leq K(p)$
- Let $M$ be a marking of a Petri Net $N$. Transition $t$, $t \in T$, can fire in marking $M$ if:
  - $p \in t_{\text{pre}} : M(p) \geq W(p,t)$
  - $p \in t_{\text{post}} : M(p) - W(p,t)$

Definition 9: Let $N = (P, T, F, W, K, M_0)$ be a $P/T$ Petri net and $M$ is a marking. If $t \in T$ is enabled in marking $M$ then it can be executed. By its execution we get new marking $M'$ which is defined as follows:

- $M(p) - W(p, t)$ if $p \in t_{\text{pre}} \setminus t_{\text{post}}$
- $M(p) + W(p, t)$ if $p \in t_{\text{post}} \setminus t_{\text{pre}}$
- $M(p) - W(p, t) + W(t, p)$ if $p \in t_{\text{pre}} \cap t_{\text{post}}$
- $M(p)$ if other.

T-transition firing from marking $M$ into marking $M'$ we denote by symbolic way $M[t] M'$.

A status of a system modelled by a Petri net is denoted by a marking of such a Net. A dynamic behaviour of a modelled system is denoted by the transition firing. Influence of a Net marking on the possibility of transition firing and, vice versa, influence of transition firing on the Net marking evolution is obvious.

Definition 10. A Non-Hierarchial Coloured Petri Net (CPN), $N_C$ is a nine-tuple $N_C = (\Sigma, P, T, A, N, C, G, E, I)$ such, so that:

- $\Sigma$ is a finite set of finite non-empty types called set of colours;
- $P$ is a finite set of places;
- $T$ is a finite set of transitions such that $P \cap T = \emptyset$;
- $A$ is a finite set of edges such that $A \cap P = A \cap T = \emptyset$;
- $N$ is a node function defined as $N : A \rightarrow (P \times T) \cup (T \times P)$;
- $C$ is a colour function defined as $C : P \rightarrow \Sigma$;
- $G$ is a transition guard function defined as $G : T \rightarrow \text{EXPR}$ such that $\forall t : T : \text{Type(G(t))} = B \land \text{Type(Var(G(t)))} \subseteq \Sigma$;
- $E$ is an arc expression function defined as $E : A \rightarrow \text{EXPR}$ such that $\forall a : A : \text{Type(E(a))} = C(p(a))_{MS} \land \text{Type(Var(E(a)))} \subseteq \Sigma$ where $p(a)$ is a place in $N(a)$, $C(x)_{MS}$ denotes that function $C$ returns for $x$ a multiset as a result;
- $I$ is an initialization function $I : P \rightarrow \text{CEXP}$ such that $\forall p : P : \text{Type(I(p))} = C(p)_{MS}$.

To complete the definition, we express that type of a variable, $v$, is denoted as $\text{Type(v)}$. If $V$ is a set of variables then $\text{Type(V)} = \{\text{Type(v)} | v \in V\}$. Let expr be an expression, then type of the expression expr is denoted as $\text{Type(expr)}$. Set of all variables inside an expression expr is denoted as $\text{Var(expr)}$. An expression expr is closed if $\text{Var(expr)} = \emptyset$. Let EXPR be a set of all well defined expressions in a used inscriptions language, then $\text{CEXP}$ is a set of all closed expressions in a used inscriptions language such that $\text{CEXP} \subseteq \text{EXPR}$. Finally, $B$ stands for set of boolean values, $B = \{\text{true}, \text{false}\}$.

Transformation Network Diagram to Timed CPN

First, we define the Timed CPN. It is a CPN extended with discrete time delay on transition firing. During such a firing the transition is blocked and capacity of target places is reduced by the expected result (colours are not affected by the time propagation; respectively, they can be fully evaluated during start of transition firing, formally).

Definition 11. A Timed CPN (T-CPN), $N_T$, is a ten-tuple $N_T = (\Sigma, P, T, A, N, C, G, E, I, \tau)$ such, so that: $N_T = (\Sigma, P, T, A, N, C, G, E, I, \tau)$ is a CPN; $\tau$ is a mapping $\tau : T \rightarrow N_0$ that assigns to every transition non-negative discrete time period denoting amount of time that must elapse during transition firing. Transitions can fire at the same conditions as in CPN a difference is in marking modification.
Starting with initial marking:

1. Set up the timer counter to zero.

2. Detect a set, $F$, of all transitions that can (possibly) fire.

3. According to previous step, denote a set, $E$, of all transitions that can fire simultaneously - in a fact, all transitions from previous detection can fire, nevertheless, if there is a conflict (for simultaneous firing) in resource sharing then conflicting transitions are resolved in a given way (according to a specified strategy, i.e. random one).

4. Split the set $E$ into two sets, $E_0$, $E_A$ such a way, so that $E = E_0 \cup E_A$, $E_0 \cap E_A = \emptyset$ and, moreover, let the set $E_0$ contains only transitions with 0 ticks time elapse during transition firing and let the set $E_A$ contains only transitions with more than 0 ticks time elapse during transition firing.

5. Start firing of all transitions in the set $E$ - start of a firing represents just removal of tokens from input places of firing transitions and, moreover, reduction of a capacity of output places (of firing transitions) according to the number of tokens that will be assigned to a given place, when the transition firing finishes.

6. Finish firing of transitions in the set $E_0$ - increase capacities of transition output places, transfer new tokens to appropriate output places (of forint transitions).

7. If there is any transition that can fire then go to the step 2.

8. Increment the timer counter by 1.

9. Detect a set, $C$, of all firing transitions a firing of which is finished in a given time period.

10. Finish firing of transitions in the set $C$ - increase capacities of transition output places, transfer new tokens to appropriate output places (of forint transitions).

11. Go to the step 2.

The own algorithm for translation of a network diagram to a T-CPN which we proposed and described in (Kvetonová, Kolář 2008), is used for next operations and modifications of the model.

An important possibility is an automated optimization of project progress based on change of input project parameters, primarily in context of real progress of a project. Especially, when any critical task is unrealizable or resource is unavailable. Another useful possibility is a reduction of redundant or unimportant information in project plan which enables us, finally, to simplify the whole Petri net model and process of its monitoring, too.

First, take into account the following input Table 1, network diagram (Figure 1) and corresponding Timed CPN model (Figure 2). On Figure 2, for the sake of readability, the resource modelling is demonstrated just for two transitions (labeled A/10 and C/12).

<table>
<thead>
<tr>
<th>ID</th>
<th>Activity</th>
<th>Dur. [days]</th>
<th>Pred.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Activity 1</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>Activity 2</td>
<td>40</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>Activity 3</td>
<td>12</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>Activity 4</td>
<td>22</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>Activity 5</td>
<td>18</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>Activity 6</td>
<td>58</td>
<td>D, E</td>
</tr>
<tr>
<td>G</td>
<td>Activity 7</td>
<td>77</td>
<td>C</td>
</tr>
<tr>
<td>H</td>
<td>Activity 8</td>
<td>36</td>
<td>C</td>
</tr>
<tr>
<td>I</td>
<td>Activity 9</td>
<td>40</td>
<td>D, E</td>
</tr>
<tr>
<td>J</td>
<td>Activity 10</td>
<td>88</td>
<td>D, E</td>
</tr>
<tr>
<td>K</td>
<td>Activity 11</td>
<td>36</td>
<td>B, F</td>
</tr>
<tr>
<td>L</td>
<td>Activity 12</td>
<td>37</td>
<td>H, I</td>
</tr>
<tr>
<td>M</td>
<td>Activity 13</td>
<td>48</td>
<td>H, I</td>
</tr>
<tr>
<td>N</td>
<td>Activity 14</td>
<td>65</td>
<td>G, M</td>
</tr>
</tbody>
</table>

Table 1: Structure of a project

Fig.1. Example of Edge-Oriented Network Diagram

MAIN RESULTS

As we can see on Figure 2, there are some small parts of the T-CPN that are unnecessarily complicated. Even if this does not matter for computer manipulation, for human driven modification/tuning of a net, this is a source of a certain disruption. Thus, we have defined and described situations, when this happens. From the description, it can be easily detected a T-CPN sub-net, which can be simplified. Moreover, we present a proof that the elimination does not influence time properties of the modified net.

Definition 12. Let $N_t$ be a Timed CPN. Let $p_0$ is a place in $N_t$ with exactly one output edge, $e_{\omega_0}$, leading to the transition, $t_\omega$. Let us extend $P$ by a new place, $p_{\omega_0}$, called a shadow place. Next, we modify $N_t$ such a way, so that $e_{\omega_0}$ is outgoing from the new place $p_{\omega_0}$ (instead from $p_0$) to the transition $t_\omega$. Let us extend $T$ by a new transition, $t_{\omega_0}$.
Fig. 2. T-CPN of a Given Edge-Oriented Network Diagram

Fig. 3. Simplified T-CPN of a Given T-CPN
Let us build a new edges, $e_{ox}$ and $e_{ix}$, going from the place $p_1$ to the transition $t_{ox}$ (edge $e_{ox}$) and the other edge going from the transition $t_{ix}$ to the place $p_{ix}$ (edge $e_{ix}$). Both edges $e_{ox}$ and $e_{ix}$ are valued by the same expression as the edge $e_{wx}$. Let the capacity of $p_{wx}$ be set to $\omega$. Let the time elapsing during transition firing for the transition $t_{wx}$ be set to $\theta$.

**Theorem 1.** By introduction of a shadow place, $p_{wx}$, the behavior of T-CPN N is not changed in the time domain (if marking of the place $p_{wx}$ is ignored).

**Proof.** The proof of Theorem 1 will be done by a contradiction. Let us assume that shadow place will modify behavior. We have to take into account the following possibilities:

1. The transition $t_{x}$ does not fire in the original T-CPN; this is because the place $p_{x}$ does not carry tokens satisfying condition for firing of the transition or for transfer via edge $e_{wx}$ in the new T-CPN, the transition $t_{wx}$ cannot fire as well, though; even if the place $p_{x}$ contains tokens satisfying valuation of edges, they would not satisfy the condition on the transition $t_{x}$; time of firing of transition $t_{wx}$ cannot be taken into account as it is zero; moreover, $p_{x}$ has the only outgoing edge, thus, any break of tokens for selection is not possible; only partial transfer (if possible) of tokens can be done in zero time development, which does not affect the overall behavior, though; which is in contradiction that the transition would fire.

2. The transition $t_{x}$ fires in the original T-CPN: in such a case, we could expect these two deviations:

(a) Transition $t_{x}$ would not fire: if a transition $t_{x}$ fires then correct tokens are stored in the place $p_{x}$ in the original T-CPN; if such a marking of $p_{x}$ holds for the new T-CPN then the transition $t_{wx}$ will fire as valuation of edges $e_{wx}$ and $e_{ox}$ is the same as for edge $e_{wx}$, the transition firing is done in a zero time elapsed; next, marking of the place $p_{wx}$ is the same as it was of the place $p_{x}$ before firing of the transition $t_{wx}$; this is exactly the same marking as for $p_{x}$ in the original T-CPN, thus, the transition $t_{wx}$ starts firing immediately; which is in contradiction with idea it will not.

(b) Transition $t_{x}$ would fire with different time elapsed: Let the transition $t_{x}$ starts firing in the time $\tau_{x}$ in the original T-CPN and let it takes $\Delta$ time units to elapse until the firing is finished; the transition $t_{x}$ starts firing in time $\tau_{x}$ in the original T-CPN then transition $t_{wx}$ starts firing in the same time in the new T-CPN as it is ready to fire exactly under the same conditions as transition $t_{wx}$ in the original T-CPN (see 2a part of this proof); taking into account definition 10, we can see that transition $t_{wx}$ finishes firing in the same time $\tau_{x}$ and, moreover, transition $t_{wx}$ starts immediately firing in the same time, which is in contradiction it would start in a different time; to assume that elapses different time amount than $\Delta$ time units during firing of $t_{wx}$, there must be a modification of the $t_{wx}$ marking/attached expressions/conditions; nevertheless, there is no such a change during introduction of a T-CPN, which is in contradiction that there is a change.

3. Token modification: this is not possible as no condition and no expression is connected with the transition $t_{wx}$ and valuation of edges $e_{wx}$ and $e_{ox}$ is the same as for edge $e_{wx}$; which is in contradiction that some modification may appear.

**Demonstration**

Result of shadow places elimination is demonstrated on Figure 3. We can see that at some places of the original net the new net is simplified. Thus, it is more appropriate for human-driven modification, as no parts of the net are representing any ballasting information. This is especially useful, if one wants to add some extra parts to model choice or repetition in the T-CPN model of a process.

**CONCLUSION**

In this paper, we have presented a new way of looking at analysis of project plans. Modelling of project plans should be done in a language, which is easy to understand and more intuitive to work with in comparison with the network diagram (especially for project planners). Moreover, analysis has to be done in more formal language similar to Petri nets. We demonstrate this by a conversion algorithm used to a transformation of a network diagram to a Timed Petri Net (T-CPN), which enables us, in the next steps, analysis of the created model.

The main goal of this approach was to combine features provided both by the network diagram (general simplicity of usage by project planners) and Petri Nets (formal machinery to model, automated control and optimization of project plans). There are two basic advantages of our approach. Firstly, we provide a user-friendly interface to define project plans (different types of activities and their dependencies) by way of a simple network diagram usage in the initial phase. Secondly, we enable the assignment of resource constraints to tasks or activities (human resources, time and/or cost) in order to enable a general optimization in resulting (transformed) Petri Net models. That means, we combine the static and dynamic aspects of project control which enable us dynamically replan the project online if necessary.

Additional research also needs to be conducted to investigate the scalability of this approach to larger systems (real projects). We can study different conflicts, too, that can appear during processes development and, thus, we can get ready to their solution in the future.

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

Demel, J. Grafi a jejich aplikace. ACADEMIA Praha, 2002.

BIOGRAPHY

ŠÁRKA KVETOŇOVÁ 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.

DUŠAN KOLÁŘ was born in Hradec Králové, Czech Republic, and went to Brno University of Technology, Czech Republic, where he studied computer science and cybernetics and obtained his degrees in 1994 and 1998. Since then, he has been working at the university, presently at the Faculty of Information Technology. His main research interests are formal languages and automata and formal models with focus on Petri Nets and reactive/triggered models.
A SERVICE-BASED ENACTMENT ENGINE FOR DYNAMICALLY RECONFIGURABLE WORKFLOWS

Franco Cicirelli, Angelo Furfaro, Libero Nigro
Laboratorio di Ingegneria del Software
Università della Calabria, DEIS, 87036 Rende (CS) – Italy
{f.cicirelli,a.furfaro}@deis.unical.it, l.nigro@unical.it

KEYWORDS
Service oriented computing, Petri nets, workflow systems, dynamic reconfiguration

ABSTRACT
This paper proposes an extension of PN-Engine, a decentralized service-based engine for the enactment of workflows, that adds support for dynamic reconfiguration. A new version of a workflow can be deployed for execution on the engine without requiring the stopping/removal of older workflow instances that are still running. Backward compatibility is ensured by a simple yet effective versioning mechanism. PN-Engine, is able to manage the execution of workflow models specified by Petri nets (PNs). The absence of a centralized control entity gives a natural support to concurrency and non determinism aspects of PNs. Advantages of dynamic reconfiguration are shown by an example concerning a workflow for a wine-production process.

INTRODUCTION
A workflow model (Hollingsworth 1995) represents a business process in a form that supports automated manipulation. A workflow defines a set of activities and a set of procedural rules that determine the specific order in which the activities must be executed to achieve a common goal. In the context of workflow management systems a challenging issue is the ability to deal with process changes (van der Aalst and Basten 2002, van der Aalst and Jablonski 2000, Qiu and Wong 2007). Process changes may be related to a case in the workflow (i.e. to a single running instance) or may involve modifications in the overall structure of the business process. The latter are referred as evolutionary changes (van der Aalst and Basten 2002) and they may result from the adoption of a new business strategy, from variations in market demand/supply, or from permanent alterations of external conditions (e.g. a change of a law).

Approaches to workflow evolution may be static or dynamic (Qiu and Wong 2007). In static solutions, once a case is started, process modifications can be made only after it has been completed whilst dynamic workflow reconfiguration does not require the stopping/removal of older cases that are still running.

To achieve workflow evolution, some different policies can be adopted (van der Aalst and Jablonski 2000):
- Forward recovery: old cases are aborted and handled outside of the workflow management system
- Backward recovery: old workflows are stopped and restarted according to the new workflow template
- Proceed: old cases are handled in the old way, new cases are handled the new way
- Transfer: old cases are transferred to the new workflow definition.

Most workflow management systems are able to support the first three policies while the fourth is still not effectively supported (Qiu and Wong 2007).

The approach proposed in this paper, which is supported by PN-Engine (Cicirelli et al. 2007c), a decentralized workflow enactment system, is novel in that it not only implements both Proceed and Transfer policies but also allows, in various degree, a mix of them.

PN-Engine is able to execute workflow models specified by PNs. It admits classic place/transition nets (P/T nets) augmented with inhibitor arcs. PN-Engine depends on the use of Service Oriented Computing (Papazoglou and Georgakopoulos 2003). Services (Bennett et al. 2000, Perrey and Lycz 2003, Cicirelli and Nigro 2007, Cicirelli et al. 2007b) are coarse-grained software components allowing a location-independent access to hardware and software resources distributed across a network.

PN-Engine, in particular, is based on Jini (Edwards and Edwards 2001, Sun Microsystems 2008) and JavaSpace (Edwards and Edwards 2001, Flennor 2001). Jini is used as the underlying service brokering infrastructure, borrowing advantages of dynamic registration, service lookup, notification of remote events, distributed object access and platform independence enabled by Java. JavaSpace is a tuple-space (Carriero and Gelernter 1990) permitting information sharing among distributed services of a Jini system. Information within a space, i.e. entries, can be read/written/taken by competing services thus implicitly synchronizing to one another. The paper demonstrates the practical use of the approach through a case study concerning a workflow for a wine-production process.

The rest of the paper is structured as follows. Next section introduces Petri nets formalism and relates it to workflow management tools and approaches. Then, details of PN-Engine architecture are illustrated. After that, issues about support of reconfigurable workflows are discussed and the chosen case study is presented. Finally, conclusions are drawn with an indication of on-going and future work.

PETRI NETS AND WORKFLOW MANAGEMENT
A Petri net (Murata, 1989), is a directed weighted bipartite graph with two types of nodes: places (graphically depicted
as circles) and transitions (graphically depicted by using a tick line). An arc connects either a place to a transition or a transition to a place. Arcs are labelled with positive integers (weights). An inhibitor arc connects a place to a transition and it can be assumed to have a weight equal to zero. An inhibitor arc is graphically represented by a dot terminated line. A marking $M$ is a function which associates a natural value to each place. A marking is graphically depicted by putting $M(p)$ black dots, called tokens, in place $p$. The state of a Petri net is defined by its marking. The initial marking is indicated with $M_0$. Formally, a Petri net is a tuple $PN = (P, T, F, W, I_{in}, M_0)$ where:

- $P = \{p_1, p_2, ..., p_m\}$ is a finite set of places,
- $T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
- $I_{in} \subseteq P \times T$ is a set of inhibitor arcs,
- $I_{in} \cap F = \emptyset$, where $\emptyset$ denotes the empty set,
- $W : F \rightarrow N$ is a weight function,
- $M_0 : P \rightarrow N$ is the initial marking,
- $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

The set of places connected to the input arcs of $t$ is referred to as the pre-set of $t$ and it is denoted by $\overset{*}{t}$. The set of places connected to the output arcs of $t$ is referred to as the post-set of $t$ and it is denoted by $\underset{*}{t}$. A transition $t$ is said to be enabled in current marking $M$ if the following precondition holds: $\forall p \in \overset{*}{t} M(p) \geq w(p, t)$ for each normal arc and $M(p) = 0$ if $(p, t) \in I_{in}$. An enabled transition may fire. The firing of an enabled transition $t$ removes $w(p, t)$ tokens from each place $p \in \overset{*}{t}$ such that $(p, t) \in F$ (withdrawal phase), and adds $w(t, p)$ tokens to each place $p \in \underset{*}{t}$ (deposit phase). Withdrawal and deposit form an atomic and instantaneous process.

Petri nets are well-suited in the domain of Workflow management (van der Aalst 1998).

A conceptual workflow specification language should at least be capable of capturing moment of choice, sequential composition, parallel execution and synchronization (van der Aalst et al. 2000). All these aspects can be naturally expressed by Petri nets. Formal semantics, local state-based system description, and abundant analysis techniques and tools make Petri nets a good choice for Workflow management systems. Petri nets can effectively be exploited for designing the workflow process definition and for workflow analysis (e.g. validation, verification and performance evaluation). Workflows are case-based, i.e. every piece of work is executed for a specific case which in turn represents a particular instance of a workflow. In the control-flow dimension, a workflow process specifies how a case is routed, i.e. which tasks need to be executed and in what order. Modelling such a process in terms of a Petri net is rather straightforward: tasks map on transitions, routing conditions can be expressed by places, and cases are modelled by tokens. The workflow state corresponds to the net marking. The control-flow dimension of a workflow is usually expressed by WF-nets (van der Aalst 1998a, Verbeek et al. 2001), i.e. classical Petri nets which satisfy two structural properties:

1. a WF-net has a source place $i$ (start condition) and a sink place $o$ (end condition), and
2. each task condition is on a path from $i$ to $o$. A WF-net is sound if it ensures proper tasks termination and avoids dead tasks.

Soundness relates to the dynamics of the net. An enabled task may require some external resources. More in general, an external condition/event can trigger task execution. For these reasons, a WF-net can be further decorated by considering triggers as tokens in additional input places of a task. A number of workflow management systems have been developed which rely on the use of Petri nets as process description language. A list of them can be found in (Verbeek et al. 2001a, van der Aalst 1998a, Guan et al. 2006). FlowManager (Aversano et al. 2002,) is a workflow management system based on coloured Petri nets. Process execution is managed by a centralized engine that, supported by a software agent layer, coordinates the distribution of subprocesses among different sites. A preprocessing step is in charge of translating the original process description into an interpretable form which is subsequently executed. A similar approach, where a Petri net process description is translated into another executable form, is adopted in Grid-Flow (Guan et al. 2006). The Grid-Flow enactment engine drives forward the control flow by coordinating the execution of programs, i.e. tasks, previously associated to transitions. Enactment engines which directly support Petri nets execution are reported in (Purvis et al. 2000), where CORBA is used as the middleware layer, and in (Verbeek et al. 2002). In the latter, a Web-based workflow management system named XRL/Flow is described. During workflow execution, the XRL/Flow engine first computes the set of enabled tasks (i.e. transitions), then dispatches it to a distribution module. By using an e-mail service, this module notifies the corresponding set of (human) resources which are responsible for tasks execution. A Web-based interface is used by a resource to signal task completion events. On receiving such an event, the engine updates tokens distribution, recomputes a new set of enabled tasks and the procedure is repeated. In (Tan and Fan 2006) a workflow model is dynamically partitioned into fragments. Each fragment can be executed on a different site. When a fragment is executed, a new set of fragments is locally determined on the basis of AND/OR split constructs, and enacted. In this way, a "push" model of control flow is obtained without requiring a centralized control entity. The approach is limited to well-structured acyclic WF-nets. While other approaches currently exist, novel in PN-Engine is the absence of a centralized enactment engine which allows net evolution to strictly adhere to Petri nets local firing semantics, i.e. net dynamic behaviour is "responsibility" of single transitions. Properties of a workflow can be proved by model checking (Cassez and Roux 2006a, Cicirelli et al. 2006). For large models, properties can be investigated by distributed simulation (Cicirelli et al. 2007a).

**PN-ENGINE**

PN-Engine (Cicirelli et al. 2007c) offers functionalities for deploying a Petri net, i.e. making it operational, and controlling net execution. Deployment of a model involves
the publication of a distinct instance of a core service, named TransitionService, for each transition existing in the model. A TransitionService is in charge of handling all the aspects related to transition firings. The marking of a deployed Petri net is stored and kept updated within JavaSpace, i.e. for each place in the net, a PlaceEntry is written into the space. Initial marking is established as part of the deployment phase. During net evolution, the management of PlaceEntry objects is responsibility of TransitionService(s). In particular, the service associated with a transition waits for receiving information about changes in its preset (i.e. events related to token deposit or withdrawal) and when the transition becomes enabled the TransitionService tries to take from the space all its preset in order to carry out the firing process. A take operation is required to ensure an exclusive access to places. This is because in a distributed scenario two or more enabled transitions might compete in acquiring the same PlaceEntry from the space. By using functionalities of the underlying middleware, a TransitionService is downloaded on the client node and executed locally. After a net is deployed, TransitionListener(s) can register themselves on TransitionService(s) in order to receive transition firing events that may trigger the execution of some activities. The approach allows decoupling net definition and deployment from the actual executors used to carry out net activities. PN-Engine supports an implicit workflow partitioning because, being based on the service metaphor, TransitionListener(s) can be operating on and belonging to different (possibly heterogeneous) cross-boundary organizations.

Fig. 1 depicts the PN-Engine architecture. PNEngineProvider offers the NetManagerService and publishes the TransitionService(s) relevant to a deployed Petri net. By using the NetManagerService, a (net)manager can feed net description into the system, deploy the net and control its execution. The ClientManagerService assists a client during the lookup of a TransitionService and the registration of a TransitionListener. During net evolution, the PNEventDispatcher service receives events due to PlaceEntry modifications within the space and dispatches token-change events informing interested TransitionService(s) about changes in net marking. Net status is mirrored into JavaSpace by using specific instances of NetStatusEntry. In this way TransitionService(s), independently from their execution locus, can be notified about status changes and can behave accordingly. On receiving a start command from the manager, a net goes into the progress status. Here, each TransitionService verifies its enabling condition and, if enabled, starts a try-to-fire operation. The try-to-fire operation is made of three phases: withdrawal, synchronous notification of firing-event and deposit. A distinct TransitionStatus entry is stored into the space for each transition and it is used to track the firing progress of the relevant transition. During the withdrawal phase the TransitionService acquires PlaceEntry objects from the space. Each place is separately taken and each time a take operation is performed, a randomly timeout is set. The timeout specifies how long a TransitionService is willing to wait for acquiring an entry already acquired by another transition. If the timeout expires, all the possibly acquired entries are released and another try-to-fire is launched. The mechanism avoids deadlock in the case two or more transitions, simultaneously enabled, share some places in their preset (conflict). When all places are acquired, a TransitionService evaluates again its enabling. Only if it finds itself still enabled, the transition may effectively fire, otherwise the try-to-fire ends by releasing the acquired entries. In the case the operation continues, the withdrawal phase terminates by decreasing the number of tokens contained in the acquired PlaceEntry objects and rewriting them into the space. The number of tokens remains zero if an inhibitor arc is involved.

Although time is not explicitly modelled, i.e. an untimed Petri net is used, the activity registered with the transition needs some time to be executed. To account for this aspect the preselection policy (Ramchandani 1974) was considered. More precisely, during withdrawal, tokens involved in a transition firing are not removed but marked as occupied (grey tokens). They will be effectively withdrawn only at the time of the deposit phase.

Figure 1: PN-Engine architecture
The behaviour of a TransitionService actually is capable also of coping with the case when multiple instances of it exist, i.e. when more listeners are registered to a single transition. In this scenario only the first among these services which gets bound to a listener, will actively participate in the transition firing, i.e. to the withdrawal and deposit phases. Remaining instances, instead, will only be informed about the transition firing. PN-Engine is able to deal with this situation making a distinction between active and passive TransitionService (see Fig. 2). Obviously the try-to-fire operation ends when all listeners complete the execution of their activities. The TransitionStatus object associated to a transition gets updated each time a TransitionService completes its activity thus allowing to determine when all activities have been carried-out.

![Figure 2: A transition having multiple listeners](image)

As a consequence of the above described try-to-fire schema, the evolution of a Petri net remains compliant with the single server semantics. In addition, each activity is not preemptable. Effectiveness of the adopted schema rests on the reasonable assumption that the time needed for solving conflicts among transitions and the time related to network delay and to the use of JavaSpace are negligible with respect to the time needed for executing the real coordinated activities.

**SUPPORTING RECONFIGURABLE WORKFLOWS**

PN-Engine supports dynamic reconfiguration of workflows by allowing selective changes in the net structure, i.e. adding/removing places, transitions and arcs. A net change usually involves more places and transitions. When the net structure has to be modified the set of variations is submitted to PN-Engine by using the NetManagerService and gets associated to an incremental version number.

In order to enable dynamic structure evolution, PN-Engine stores version information along with the history of structural changes maintaining for each transition a specific JavaSpace entry named TransitionInfo. Version information is also stored inside the objects representing the tokens, i.e. PlaceEntry(s).

A TransitionInfo owns a list which takes track of the history of variations in the preset and posset of the relevant transition t. A structure change, involving t, i.e. t is added to an existing net or its preset and posset relations are modified, causes the addition of a new item into this list. Each of these items stores the version number of the net, the new preset and posset and a firing strategy.

A firing strategy is in charge of determining which version of the transition has to be used on the basis of the version of tokens contained in the places that are, or were, part of its preset and of associating a version to the tokens it generates. Structural changes are atomically actualized by exploiting transactional functionalities of JavaSpace which in turn notifies the relevant TransitionService(s).

The firing strategy may be chosen from a standard set offered by PN-Engine or custom defined by the user. In the following some strategies are briefly surveyed.

The Update strategy always fires the most recent version of the transition, does not care of the token versions of its actual preset for evaluating its enabling, transfers to the token it generates the current net version. This strategy gives support to the Transfer policy (van der Aalst and Jablonski 2000).

The Propagate strategy is a variant of Update in that it transfers to the generated tokens the most recent version among those of the consumed tokens.

The Select strategy selects the transition version that has to be used on the basis of the version of the token contained into a user-specified place. This version is transferred to the generated tokens.

The Retain firing strategy, each time it detects a change in one of its historical presets, searches, starting from the most recent, for a version of the transition that is enabled by tokens of the same version. If such a version is found, it is fired and to the generated tokens is given the same version of those consumed. This strategy supports the Proceed policy (van der Aalst and Jablonski 2000).

![Figure 3: Retain strategy example](image)

Fig. 3(a) and Fig. 3(b) depict two different versions, namely $v_1$ and $v_2$, of transition $t$. In Fig. 3(a) the activity corresponding to $t$ needs only resource A whose availability is modelled by the marking of place $P_a$. In Fig. 3(b) the same activity now requires also an instance of resource B (place $P_b$). Let us suppose that place $P_c$ contains one token, modelling a specific workflow case, and that transition $t$ adopts the Retain strategy. The enabling of $t$ depends on the version of the token in $P_c$. If this version is $v_1$, $t$ requires only a token in $P_a$ (of version $v_1$), otherwise it requires a token in $P_b$ and another in $P_a$ both having version $v_2$.

If carefully exploited, firing strategies foster backward compatibility of different versions of the same workflow that can coexist and evolve simultaneously.

Another important feature of PN-Engine is the support for version rollback, i.e. the capability of restoring a particular workflow version. Rollback to version $v_m$ is naturally carried-out by ignoring all the items inside TransitionInfo(s) having a version number greater than $v_m$. Rollbacks do not
physically remove any information and then are easily undoable.

**A WINE-PRODUCTION PROCESS**

This section illustrates a usage of dynamic workflow reconfiguration, as supported by PN-engine, by considering an example of a wine-production process in the context of a small winery. The aim of this winery is the production of high-quality wine made from vintages of local vineyards or from musts (i.e. grape juices) purchased from quality-certified vineyards. The wine-production process is made of the following phases:

- **Grape harvest.** Grapes are picked from the vineyard and carefully gathered to the factory into proper containers. Then, they are put inside suitable holding-tanks.
- **Destemming and crushing.** Destemming is the process of separating stems from the grapes. Crushing is the process of gently squeezing the berries and breaking the skins to start liberating the contents of the berries.
- **Enzyme treatment.** During this phase, which lasts from 2 to 4 hours, some enzymes are added to the must by a floating unit.
- **Filtration.** All the grape skins are separated from the must by filters or centrifuges before the must undergoes fermentation.
- **Fermentation.** The must is pumped to a fermentation container where it settles for a few days.
- **Ageing.** The wine is tightly packed in oak barrels not allowing the air to enter in them for nearly several months.
- **Bottling.** After the ageing phase is completed, the wine is transferred into coloured glass bottles.
- **Labelling.** The bottles are labelled and stored into warehouses waiting to be sorted.

The fulfilment of the above phases requires various resources most of which are of physical nature, e.g. holding-tanks, floating units, oak bottles and so on. Also some human resources are needed in the production process and are involved in different phases. Three types of human resources are identified:

- **Administrative agent (AA).** He is in charge of monitoring the arrival of grapes and musts and the storing of bottles into warehouses.
- **Quality checker (QC).** He is responsible for controlling and warranting the quality of the phases from the arrival of raw material to the wine fermentation and for the quality assessment at the end of the ageing phase.
- **Production operator (PO).** He is involved in all the operations that require the intervention of a human operator.

The bold part of net shown in Fig. 4 is the model of the workflow for the described wine-production process. The grey part is related to some additions made to cope with a process evolution as described in the next section.

Places with dashed borders are references to the actual places having the same names and are used to increase model readability.

The process begins when harvested grapes or musts reach the factory. Places **AA, QC and PO** correspond to the above-described human resources. A token in place **PG1 or PM1** respectively models the arrival of grapes or must. Transitions from **TG1 to TG6** correspond to the sequence of phases from grape unloading to filtration. Before a new grape delivery these phases must have been completed. This is ensured by place **MTX1** used for mutual exclusion. Place **MTX2** handles must deliveries in an analogous way. Place **PM** gathers musts ready for the ageing phase. Transitions from **TB1 to TB5** and from **TB6 to TB10** respectively model the sequences of wine loading/maturity/unloading in the two available oak barrels. Following maturation, both barrel cleaning (**TBC1 and TBC3**) and wine bottling (**TB11**) are started in parallel. Labelling (**TL1**) and warehousing (**TW1 and TW2**) phases complete the workflow. In this workflow all the transitions are equipped with the **Propagate** firing strategy. It should be noted that the model in Fig. 4 is not a WF-net but it could be easily turned into a WF-net, if required.

**Workflow evolution**

Let us now suppose that the winery wants to expand its market by adding to its business also the production of medium quality wines. Whilst the production process is almost the same, some new and different physical resources are needed. For example stainless-steel tanks instead of oak barrels will be used for the ageing of medium quality wines. The process workflow has to be modified accordingly for taking into account the new type of production, which must coexist with the previous one, and the newly available physical resources. The available human resources remain the same.

After modifications, the Petri net model is that of Fig. 4 where also the grey parts are considered. The new workflow needs to distinguish whether the raw material is quality-certified or not and, based on this information, it has to discriminate the processing phases that have to be performed. The marking of places **PC1 and PC2** are respectively used to indicate whether the grapes or the must are quality-certified. Before the beginning of the ageing phase, the workflow is now able to decide if the must has to be transferred into one of the oak barrels or into one of the stainless-steel tanks based on the quality certification. After the ageing is completed the bottling phase is the same as before and it is common for both types of wine. However, the quality information is used again (place **PWC**) during the labelling and warehousing phases.

When the workflow reconfiguration is performed, a process that already begun with the old version may have not yet completed. For example, musts that were in the ageing phase at reconfiguration time have to be treated as quality-certified even with no certification (which was not required in the first workflow version). A key point for enabling backward compatibility, i.e. ensuring that such a process is consistently handled, relies on a careful adoption of the transition firing strategies.
Figure 4: The wine-production workflow and its resources
Table 1 lists transitions adopting the Select strategy. Some of them were added to the new workflow version while others were already present but had been modified. All the other added transitions were equipped with the Propagate strategy.

<table>
<thead>
<tr>
<th>Transitions</th>
<th>Selection Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGl and TGlh</td>
<td>Grapes</td>
</tr>
<tr>
<td>TGl6 and TGlh6</td>
<td>PG6</td>
</tr>
<tr>
<td>TM1 and TM1h</td>
<td>Must</td>
</tr>
<tr>
<td>TM3 and TM3h</td>
<td>PM3</td>
</tr>
<tr>
<td>TB11</td>
<td>PB10</td>
</tr>
<tr>
<td>TL</td>
<td>Bottles</td>
</tr>
<tr>
<td>TW1</td>
<td>Pw1</td>
</tr>
</tbody>
</table>

**Workflow deployment and execution**

The proposed model was executed on four Win platforms Pentium IV 3.4GHz interconnected by a 1GB Ethernet switch. One computing node was used as provider for the PN-Engine services and as provider for the required Jini services and JavaSpace. Remaining nodes were used for hosting TransitionListener(s). In a real scenario TransitionListener(s) could be deployed accordingly to the actual placement of the resources within the winery context. Three TransitionListener(s) were considered respectively for the administrative agent (AA), quality checker (QC) and production operator (PO).

The net administrator deploys the workflow by submitting a file containing the net description to the NetManagerService by exploiting its GUI (see Fig. 5). On the basis of such information, the NetManagerService automatically writes the relevant data structures into the JavaSpace and then instantiates and publishes the various TransitionService(s). Subsequently, each listener looks up for the TransitionService(s) relevant to the transitions it has to listen to, then it registers on them and stays at rest until notified by a firing event. This phase is carried out by using the ClientManagerService (see Fig. 6) which assists lookup and registration steps. In the cases where a listener is tied to a human resource, it acts as a notifier that, upon receiving a firing event, highlights the activity which has to be performed in the GUI used by the human operator.

Fig. 7 shows the GUI associated to the production operator as it appears in the first workflow version. After a workflow evolution, the set of activities performed by a human resource may change. As a consequence the relevant listeners must change their set of registered transitions. This is achieved by exploiting again the functionalities of the ClientManager service. These modifications are mirrored in the activity list of the human operator’s GUI (see Fig. 8).
CONCLUSIONS

This paper describes an approach to dynamic reconfiguration of workflows specified by Petri nets and centred on an extension of PN-Engine (Cicirelli et al. 2007c). Reconfigurability relies on a versioning mechanism that allows changes in the net structure and enables the coexistence of multiple versions of a workflow. The absence of a central coordination entity allows to delegate the handling of versioning concerns to each single transition. Compatibility of workflow cases, belonging to different versions, is made possible by a customizable firing strategy associated to transitions. A firing strategy determines the way a transition behaves depending on the versions of the tokens contained in places that are, or were part, of its preset.

On-going work is geared at:

- implementing a data-exchange mechanism among TransitionListener(s), e.g. by attaching data to tokens in the space
- using PNML (Billington et al., 2003), an XML based interchange format, for representing Petri net models in a portable way and exploiting its modular constructs for dealing with complex models
- introducing explicit time management within a modellled system by supporting a timed version of Petri nets
- exploiting PN-Engine for the orchestration of composed services.

REFERENCES


PETRI NETS MODELS
ALGORITHMS FOR COMPUTING COVERABILITY GRAPHS FOR CONTINUOUS PETRI NETS

Petr Novosad and 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

Continuous Petri Net, Coverability graph, Algorithm.

ABSTRACT

A useful method for Petri nets analysis is based on a coverability tree or a coverability graph that represents a state space of a system. The paper formalizes the concept of the coverability graphs for autonomous continuous Petri nets and introduces algorithms for their computing. The algorithms for the bounded and unbounded continuous Petri nets are presented. Every algorithm is accompanied with an example. The new concept of a signed continuous macro-marking for the bounded continuous Petri nets is described.

INTRODUCTION

Petri nets are a mathematical and graphical tool for modeling concurrent, parallel and/or distributed systems. A finite reachability tree is a useful structure for checking properties of the Petri nets. Our goal it to update algorithms for computing the coverability graphs for the discrete Petri nets to the continuous Petri nets.

The article consists of the following. The definitions and notations of the continuous Petri nets are given in Section 2. Section 3 presents the algorithms for the coverability graphs with examples. Section 4 concludes the paper.

CONTINUOUS PETRI NETS

The concept of the continuous Petri nets has been presented by David and Alla in 1987 (David and Alla 1987; David and Alla 1998; David and Alla 2005). It is a fluidification of the discrete Petri net. Places can hold a real valued marking. This paper assumes that the reader is familiar with the basic theory of the Petri nets (Reisig 1985; Desel and Juhás 2001). The Petri net is bounded if the number of tokens in all places and in all reachable markings is less than some upper bound.

Continuous Petri net (David and Alla 2005) is defined as a 5-tuple $R = (P, T, Pre, Post, M_0)$, where $P$ is a finite set of places and $T$ is a finite set of transitions. $P \neq \emptyset$, $T \neq \emptyset$ and $P \cap T = \emptyset$. Pre: $P \times T \rightarrow Q^+$ is the input incidence matrix. Post: $P \times T \rightarrow Q^+$ is the output incidence matrix. $M_0: P \rightarrow \mathbb{R}^+$ is the initial marking (Notation $Q^+$ corresponds to the non-negative rational numbers and notation $\mathbb{R}^+$ corresponds to the non-negative real numbers, both including zero). For $p \in P$ and $t \in T$, $Pre(p, t)$ is the weight of the arc $p \rightarrow t$, $Post(p, t)$ is the weight of the arc $t \rightarrow p$. If the arc does not exist, the weight is 0. A graphical representation of the continuous Petri net is shown in Figure 1.

A continuous marking $m \in (\mathbb{R}^+)^{|P|}$ is a vector of non-negative real numbers. A transition $t \in T$ is enabled in the marking $m$ if $\forall p \in \mathbb{R}^+$ such that $m(p) > 0$. Enabling of a transition does not depend on the arc weight, it is sufficient that every input place has a non-zero marking. An enabling degree $q$ of the transition $t$ for the marking $m$ is the maximal amount that the transition can fire in one go, i.e. $q(t, m) = \min_{p \in \mathbb{R}^+} (m(p)/Pre(p, t))$. Firing the transition $t$ with a quantity $\alpha < q(t, m)$ the marking $m'$ is denoted as $m \leftarrow [t]^{\alpha}$, where $[t]^{\alpha}$ represents $\alpha \in \mathbb{R}^+$ firings of the transition $t$ at one go (according to notation in (David and Alla 2003)). The new marking $m' = m + \alpha C(P, t)$, where $C = Post - Pre$ is a token-flow matrix. The marking $m'$ is reachable from the marking $m$. The set $P$ of places may be divided into two subsets: $P^+(m)$ the set of the places $p \in P$ such that $m(p) > 0$, and the set of the places $p$ such that $m(p) = 0$. A continuous macro-marking is the union of all markings $m$ with the same set $P^+(m)$ of the marked places. Since each continuous macro-marking is based on the Boolean state of every place (marked or not marked), the number of the continuous macro-markings is less than or equal to $2^n$, where $n$ is the number of the places. Note: for shortening, the macro-marking refers to the continuous macro-marking in the following text.

COVERABILITY GRAPHS

For the discrete Petri nets there exists an algorithm developed by Karp and Miller (Karp and Miller 1969) for computing a reachability root tree. The state space of the unbounded Petri net is there presented as a finite tree. The coverability graph gathers nodes which correspond to the same marking (Coves et al. 1998) using a symbol $\omega$ that represents arbitrarily many tokens. Nodes in the coverability graph represent states of a system and edges represent transition firings. There exist algorithms for checking qualitative properties such as safeness, boundness, conservativeness, coverability and reachability for the bounded Petri nets from the coverability graphs. However one cannot check properties such as deadlock, liveliness and reachability for the unbounded Petri nets because of the aggregation of markings and thus the loss of information.
Coverability Graphs For Bounded Continuous Petri Nets

The concept of the coverability graph for the continuous Petri nets was introduced in (David and Alla 2003), without a formalization and an algorithm. Because of the continuous change of the marking, even the bounded continuous Petri net has an infinite state space, thus the reachability graph cannot be constructed and analysis options are limited.

We define a coverability graph $G_{cb}$ for the bounded continuous Petri net $R = (P, T, Pre, Post, M_0)$ as a pair $G_{cb} = (N, E)$. The set $N \subseteq (\mathbb{R}^+ \cup \{c_i, i = 1 \ldots |P|\})^{|P|}$ is the set of states, where $c_i$, $i = 1 \ldots |P|$ is a substitute symbol that represents non-zero marking in the continuous place $p_i$. The elements of $N$ are the macro-markings. The set $E \subseteq N \times T \times ((\mathbb{R}^+ \setminus \{0\}) \cup \{c_1, \ldots, c_{|P|}\}) \times N$ is the set of edges labeled with a fired transition and its degree. The degree can be substituted with the symbol $c_i$. Firing a transition without specifying the firing quantity means that it was fired with less quantity than the maximal enabling degree.

The algorithm 1 constructs the coverability graph for the bounded continuous Petri net. A function AddNewNode() adds a new node for the given marking and flags it as unprocessed. A function GetEnabledTransitions() returns a set of enabled transitions for the macro-marking represented by the given node. A function GetEnablingDegrees() returns a set of enabling degrees that are valid for the given transition and node. It returns values of a maximal enabling degree and a half of the maximal enabling degree. A function GetNode() returns an existing node for the given marking. A function FireTransition() returns a new macro-marking after firing the given transition with the given degree in the macro-marking represented by the given node. The new macro-marking $m$ is created as follows. If a marking in the place $p_i$ is a boundary value (including zero) then the marking $m(p_i)$ is set to this value. Else the marking $m(p_i)$ is set to the substitute symbol $c_i$. The substitute symbol $c_i$ for the place $p_i$ need not to propagate to the following marking. Basically, new nodes are created when a marking of a place becomes zero, or when an unmarked place becomes marked.

An example of the coverability graph created by the algorithm 1 for the bounded continuous Petri net in Figure 1 is shown in Figure 2. The net has two places, therefore it can has a maximum of 4 macro-markings. Reachable macro-markings are $(1, 0), (c_1, c_2)$ and $(0, 1)$.

**Algorithm 1**: The coverability graph for the bounded continuous Petri nets

**Input**: The bounded continuous Petri net $R = (P, T, Pre, Post, M_0)$

**Output**: The coverability graph $G_{cb} = (N, E)$

**Method**:

begin

AddNewNode $(M_0)$;

while exists a node $n \in N$ such that $n$ is unprocessed do

flags the node $n$ as processed:

$F = \text{GetEnabledTransitions}(n)$;

for each transition $t \in F$ do

$Q = \text{GetEnablingDegrees}(n, t)$;

for each degree $q \in Q$ do

$m' = \text{FireTransition}(n, t, q)$;

if a node with $m'$ does not exist in $N$ then

AddNewNode $(m')$;

end

$n' = \text{GetNode}(m')$;

if an edge $(n, t, q, n')$ does not exist in $E$ then

AddNewEdge $(n, t, q, n')$;

end

end

end

end

![Figure 1: Bounded Continuous Petri Net](image1.png)

![Figure 2: Coverability Graph](image2.png)
Algorithm 2: The extended coverability graph for the bounded continuous Petri nets

**Input:** The bounded continuous Petri net $R = (P, T, \text{Pre}, \text{Post}, M_0)$

**Output:** The extended coverability graph $G_{ecb} = (N, E)$

**Method:**
This algorithm is similar to the algorithm 1. The main difference is in the function $\text{FireTransition}()$, that can return the new signed type of the macro-markings. For the given edge $t$, the signs for the symbols $c_i$ in the places $p_i$ are set according to the token-flow matrix (whether $C(p_i, t)$ is positive or negative).

![Figure 3: Extended Coverability Graph](image)

Coverability Graphs For Unbounded Continuous Petri Nets

The coverability graph for the unbounded continuous Petri net is a pair $G_{cu} = (N, E)$, where $N \subseteq (\mathbb{R}^+ \cup \{\omega\} \cup \{c_1, \ldots, c_p\})^{|P|}$. The set $E$ is the same as for $G_{cb}$. The symbol $\omega$ represents an arbitrarily large number in a place and the marking in a such place is unbounded.

Algorithm 3: The coverability graphs for the unbounded continuous Petri nets

**Input:** The unbounded continuous Petri net $R = (P, T, \text{Pre}, \text{Post}, M_0)$

**Output:** The coverability graph $G_{cu} = (N, E)$

**Method:**
This algorithm is similar to the algorithm 1. The main difference is in the function $\text{FireTransition}()$, that can return the macro-markings with the symbol $\omega$. The macro-marking $m''$ depends on the previous macro-marking $m'$ as follows. If the macro-marking $m''$ covers the macro-marking $m'$ ($m''(p_i) > m'(p_i)$ then for places where $m''(p_i) > m'(p_i)$ set $m'(p_i) = \omega$. The symbol $\omega$ in the place $p_i$ must propagate to the succeeding macro-markings in the place $p_i$.

An example of the coverability graph created by the algorithm 3 for the unbounded continuous Petri net in Figure 4 is in Figure 5. The system in Figure 4 has a self-loop and the place $P_2$ is unbounded. This is projected to the macro-markings $(1, \omega, 0), (0, \omega, 1)$ and $(c_1, \omega, c_3)$ in Figure 5.

![Figure 4: Unbounded Continuous Petri Net](image)

![Figure 5: Coverability Graph](image)

**CONCLUSION AND FUTURE WORK**

In this paper, we have introduced the definitions of the coverability graphs for the autonomous bounded and unbounded continuous Petri nets. We have shown the corresponding algorithms for their computation and we are currently implementing them. Future work lays mainly in the extension of the algorithms to hybrid Petri nets.

*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**


CONTINUOUS APPROXIMATION OF PEPA MODELS AND PETRI NETS

Vashti Galpin
LFCS, School of Informatics, University of Edinburgh, Scotland
Email: Vashti.Galpin@ed.ac.uk

KEYWORDS
PEPA, Petri nets, continuous approximation, state space explosion, server semantics

ABSTRACT

Modelling very large systems that consist of many similar components can lead to state space explosion. Continuous approximation can avoid this. In the stochastic process algebra PEPA, models with large numbers of identical components can be approximated in a continuous fashion by a set of coupled ordinary differential equations (ODEs). Similarly, timed continuous Petri nets can be used to approximate behaviour via ODEs where there are many servers. These two approaches are compared and infinite and finite server semantics are considered.

INTRODUCTION

When modelling large systems such as one providing Web services to many clients, state space explosion imposes restrictions on the size of system that can be analysed. An approximation approach can mitigate the state space explosion problem by making it unnecessary to construct the state space. Ordinary differential equations (ODEs) extracted from the stochastic process algebra model of the Web services system were analysed in less than a tenth of a second of compute time even though there were $3^N$ states for $N$ clients (Gilmore and Tribastone 2006).

The continuous approximation technique for the stochastic process algebra PEPA (Hillston 1996) provides coupled ODEs which model changes in behaviour over time (Hillston 2005). This paper compares this approach with timed continuous Petri nets (Mahulea et al. 2006, David and Alla 2005) and the associated ODEs. In both cases, the continuous approximation deals with the problem of state space explosion. For PEPA, the state space is the labelled multi-transition system obtained from the structured operational semantics and in (discrete) bounded stochastic Petri nets (Ajmone Marsan et al. 1995), it is the reachability graph which describes the possible markings of a Petri net. For both PEPA and stochastic Petri nets, if the rates of activities/transitions are exponentially distributed, then the transition system/reachability graph is the basis for a continuous time Markov chain (CTMC) that represents the behaviour of the system over time (Hillston et al. 2001). The continuous approximations avoid the derivation of the state space, and the models become amenable to analysis (Hillston 2005).

In timed-based Petri nets, nets have either infinite server semantics or finite server semantics (multiple or single) (Mahulea et al. 2006). In infinite server semantics as many simultaneous firings of a transition as are enabled can take place. In finite server semantics, there is a finite bound on this number. These represent two different approaches to coordinating the interactions of components. The two semantics will be compared.

In this paper, a translation from PEPA models to continuous Petri nets and vice versa will be presented and it will be shown that the ODEs for each approach are the same. Additionally, it will be shown that the continuous semantics for PEPA are infinite server semantics. These results allow the use of techniques and theory for one approach to be applied to the other, and offer a choice between two formalisms to represent systems, either graphical or textual.

This research is novel. PEPA and stochastic Petri nets have been compared (Hillston et al. 2001, Ribando 1995) but the continuous approximations have not. Furthermore no investigation has been conducted as to whether the continuous approximation for PEPA has finite or infinite server semantics. These are important to consider as because they deepen our understanding of the two modelling formalisms.

In the next section, continuous Petri nets are introduced, after which a section follows on the ODE semantics of PEPA. The fourth and fifth sections of the paper provide the translations, followed by a discussion of server semantics. The final section covers conclusions and further work.

CONTINUOUS PETRI NETS

This section presents existing definitions (Mahulea et al. 2006, Hillston et al. 2001). Let $\mathbb{R}^+ = \{x \mid x \geq 0\}$.

Definition 1 A (timed) continuous Petri net (CPN) is a pair $(N, m_0)$ where $N = (P, T, Pre, Post, \lambda)$ with $P$ the set of places, $T$ the set of transitions with $P \cap T = \emptyset$, and $Pre : P \times T \rightarrow \mathbb{R}^+$ and $Post : P \times T \rightarrow \mathbb{R}^+$ are the pre and post incidence matrices respectively which give the arc weights between places and transitions. A net is ordinary if all arc weights have value 0 or 1. The token flow matrix is $C = Post - Pre$. Additionally for $t \in T$, $t^* = \{p \mid Pre(p, t) > 0\}$, $t^+ = \{p \mid Post(p, t) > 0\}$, and for $p \in P$, $p^* = \{t \mid Post(p, t) > 0\}$, $p^+ = \{t \mid Pre(p, t) > 0\}$. The function $\lambda : T \rightarrow \mathbb{R}^+$ associates with each transition a firing rate. A marking associates values with places at a specific point in time, and is defined as a function $m : P \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$. The initial marking is $m_0 = m(\cdot, 0)$. 

492
CPNs are drawn in the standard manner using circles for
places and rectangles for transitions as illustrated in
Figure 1 which describes a system with 200 clients and 100
unreliable servers that fail and then are repaired. The clients
prepare work, obtain a service from a server and then
finish the work, before starting on the next job. The initial
marking is \((200, 0, 0, 100, 0, 0)\). The transition firing rates
are \(\lambda(prepare) = p\), \(\lambda(server) = s\), \(\lambda(finish) = f\),
\(\lambda/reset) = r\), \(\lambda(fail) = a\) and \(\lambda(repair) = e\).
The distinguishing feature of CPNs is that markings are
not restricted to integer values. In this paper ordinary nets
are used. Hence, whenever \(p \in t^*\) then \(Pre(p, t) = 1\) and
whenever \(p \in t\) then \(Post(p, t) = 1\). Results can be generalised
since a net with arc weights greater than one can be
converted to an ordinary Petri net (David and Alla 2005).

**Definition 2** A transition \(t\) is enabled at time \(\tau\) if for all \(p \in t, m(p, \tau) > 0\) and \(t\) has enabling degree

\[
\text{enab}(t, \tau) = \min_{p \in t^*} \left\{ \frac{m(p, \tau)}{Pre(p, t)} \right\}
\]

An enabled transition can fire an amount of \(\alpha\) with \(0 < \alpha \leq \text{enab}(t, \tau)\). After this firing the new marking will be

\[
m(\cdot, \tau') = m(\cdot, \tau) + \alpha C(\cdot, t).
\]

Mahulea et al. (Mahulea et al. 2006) note that the continu-
ous approximation under infinite server semantics is appro-
priate for many clients and many servers. A deterministic
continuous approximation of the discrete case can be done
by assuming that the firing delays can be approximated by
their mean values (Silva and Recalde 2005). This applies to
mono-T-semiflow reducible nets which includes the class of
equal conflict nets (Mahulea et al. 2006).

The fundamental equations which defines how markings
change over time is defined as

\[
m(\cdot, \tau) = m_0 + C \sigma(\tau) \quad \text{where} \quad \sigma(\tau) \text{ is the firing vector (Mahulea et al. 2006)}.\]

Differentiating this equation with respect to time gives

\[
\frac{dm(\cdot, \tau)}{d\tau} = Cf(\cdot, \tau)
\]

with

\[
f(t, \tau) = \lambda(t) \min_{p \in t^*} \left\{ \frac{m(p, \tau)}{Pre(p, t)} \right\} = \lambda(t) \min_{p \in t^*} \left\{ m(p, \tau) \right\}
\]

where the flow \(f(\cdot, \tau) = \sigma'(\tau)\) is the derivative (in the math-
ematical sense) of the firing sequence (Mahulea et al. 2006).
The change in the marking of a single place \(p\) can be ex-
pressed as follows. Let \(n\) be the number of transitions. Then

\[
\frac{dm(p, \tau)}{d\tau} = \sum_{j=1}^{n} \text{Post}(p, t_j) f(t_j, \tau)
\]

\[
- \sum_{j=1}^{n} \text{Pre}(p, t_j) f(t_j, \tau)
\]

\[
= \sum_{\tau \in p} f(t, \tau) - \sum_{\tau \in p^*} f(t, \tau)
\]

\[
= \sum_{\tau \in p^*} \lambda(t) \min_{\tau' \in p^*} \left\{ m(p', \tau) \right\}
\]

\[
- \sum_{\tau \in p^*} \lambda(t) \min_{\tau' \in p^*} \left\{ m(p', \tau) \right\}
\]

**PEPA AND ODE SEMANTICS**

Consider the standard syntax for PEPA (Hillston 2005) with
Hiding omitted: \(P ::= P \bowtie P \mid C \text{ and } S ::= (\alpha, r, S) \mid S + S \mid C_s\) where \(C\) names a model or sequen-
tial component, \(C_s\) names a sequential component, \(\alpha\) is an action, \(r\) is a rate, and
and they form an activity, and \(L\) is a set of actions. Also
consider the standard operational semantics of PEPA given
in Figure 2 (Hillston 1996) with the following operators.

**Prefix** \((\alpha, r).P\) can be understood as the process that can
perform the action \(\alpha\) with a delay from the exponential dis-
tribution determined by \(r\) and which then behaves as \(P\).

**Choice** \(P_1 + P_2\) represents the choice between behaving as
either \(P_1\) or \(P_2\). The process that completes first will proceed
and the other will be discarded.

**Cooperation** \(P_1 \bowtie P_2\) can act as \(P_1\) independently of \(P_2\)
(and vice versa) for any actions not in \(L\). For actions in
\(L\), \(P_1\) and \(P_2\) can only proceed when they both can perform
the action and the rate is determined by the slower of the two.
\(P_1||P_2\) is used for \(P_1 \bowtie P_2\), and \(P[n]\) for \(n\) copies of \(P\) in
parallel without cooperation.

**Constant** \(C \equiv P\) defines the constant \(C\) which has the same
behaviour as \(P\).

The following subset of PEPA will be considered, as this is
the subset for which the ODE semantics are defined (Hillston
2005). It only allows communication between components
that are not identical.

1. All shared actions of non-identical components must syn-
chronise.
2. Identical components cannot synchronise on actions.
3. Rates must be identical for a shared activity, and no
passive rates are allowed.

**Definition 3 (Hillston 2005)** For an arbitrary PEPA model
\(M\) with \(n\) component types \(C_i\) for \(i = 1, \ldots, n\) each with
\(N\) distinct derivatives (successor states of components), the
numerical vector form of \(M\), \(V(M)\) is a vector with \(N = \sum_{i=1}^{n} N_i\) entries. Each \(v_{ij}\) records how many instances of
the jth local derivative of component type $C_i$ are present in the current state.

The numerical vector form can be used to obtain the vector state space which is smaller than the labelled multi-transition system when there are many identical components (Hillston 2005). The vector state space gives an aggregated model and can generate a CTMC. The numerical vector form is used for the continuous approximation in the case of large numbers of identical components using the following differential equation for $N(C_i, \tau) = v_{ij}$, the number of derivatives of type $C_i$ at time $\tau$ (Hillston 2005).

$$\frac{dN(C_i, \tau)}{d\tau} = \sum_{(\alpha, r) \in En(C_i)} \min_{C \in Ex(\alpha, r)} \{N(C, \tau)\}$$

for $Ex(D) = \{((\alpha, r) \mid D^{(\alpha, r)D})\}$, $Ex(\alpha, r) = \{D \mid D^{(\alpha, r)}\}$ and $En(D) = \{((\alpha, r) \mid D^{(\alpha, r)}\}$. Hence $Ex(D)$ captures those activities that decrease the number of $D$'s (exit activities), $En(D)$ captures those activities that increase the number of $D$'s (entry activities) and $Ex(\alpha, r)$ describes those derivatives that can perform $(\alpha, r)$ activities. Therefore the change in the number of a derivative $D$ is expressed in terms of the number of decreases and increases at the given rate where the changes are bounded by the minimum number of derivatives available to perform the activity. Extracting the specific ODEs for a given model can be automated based on an activity graph or an activity matrix (Hillston 2005).

**Definition 4** An activity graph is a bipartite graph $(N_A)$. The nodes are partitioned into $N_i$, the activities, and $N_p$, the derivatives. $A \subseteq (N_i \times N_p) \cup (N_p \times N_i)$, where $a = (n_i, n_p) \in A$ if $n_i$ is an entry activity of derivative $n_p$, and $a = (n_p, n_i) \in A$ if $n_i$ is an exit activity of $n_p$. The activity matrix $M_a$ is an $N_p \times N_i$ matrix and entries are defined as follows.

$$M_a(p_i, t_j) = \begin{cases} +1 & \text{if } t_j \in En(p_i) \\ -1 & \text{if } t_j \in Ex(p_i) \\ 0 & \text{otherwise.} \end{cases}$$

The example in Figure 3 consists of two different types of servers, possibly offering the same service, but at different rates, plus clients who do not mind which server they use. The servers provide the service then reset, and a client interacts with a server and then does something before interacting with a server again. The activity matrix is given in Figure 5 together with the ODEs for this model. The activity graph for this system is given in Figure 4.

**FROM A PEPA MODEL TO A CPN**

Working with the subset of PEPA specified previously, it is possible to convert this to a timed CPN. First, construct the activity graph. This is a net of the form $(N_p, N_i, M_{pre}, M_{post})$ where

$$M_{pre}(p_i, t_j) = \begin{cases} +1 & \text{if } t_j \in Ex(p_i) \\ 0 & \text{otherwise.} \end{cases}$$

$$M_{post}(p_i, t_j) = \begin{cases} +1 & \text{if } t_j \in En(p_i) \\ 0 & \text{otherwise.} \end{cases}$$

The activity matrix $M_a$ is then $M_{post} - M_{pre}$ which is the token flow matrix. The initial marking is determined by the numbers of each component; $m(p, 0) = N(p, 0)$. Additionally, $\lambda(t) = r$ where $t = (\alpha, r) \in N_i$. Hence a CPN can be constructed from a PEPA model, and the activity graph and the CPN are isomorphic.

Moreover, for each distinct type of derivative in the model, there is a place in the Petri net and the tokens in that place represent the number of copies of the derivative in the model. The vector state space obtained from the PEPA model is isomorphic to the reachability graph of the CPN if it were viewed as a discrete Petri net, since the behaviour of both the PEPA model and the net is the same. If an activity reduces by one the count of a particular derivative and increases by one the count of another derivative in the PEPA model, then in the Petri net, the transition that is that activity will fire and remove one token from the place that represents that particular derivative and add a token to the place that represents the other derivative. Similarly the token changes associated
\[
C \equiv (\text{serv}_1, s_1).C' + (\text{serv}_2, s_2).C' \quad C' \equiv (d_0, d).C' \quad S_i' \equiv (\text{reset}_i, r_i).S_i \quad S_i \equiv (\text{serv}_1, s_i).S_i' \quad \text{Sys} \equiv C'[100] \quad \bigotimes \bigotimes \bigotimes \bigotimes (S_1[50][||S_2[50]]
\]

Figure 3: Servers Example

![Diagram](https://via.placeholder.com/150)

Figure 4: The Activity Graph and its Translation to a CPN with Initial Marking for the Servers Example

with a firing of a transition can be expressed as changes in the number of derivatives in the PEPA model. Hence, \( m(p, \tau) = N(p, \tau) \). The resulting net is bounded because the total number of derivatives for each component is fixed in the PEPA model.

Considering the continuous case, the ODEs defined from PEPA for the place \( p \) are as follows.

\[
\frac{dN(p, \tau)}{dt} = \sum_{t \in En(p)} r \times \min_{p' \in Ex(t)} \{N(p', \tau)\} - \sum_{t \in Ex(p)} r \times \min_{p' \in Ex(t)} \{N(p', \tau)\} = \sum_{t \in *, p} \lambda(t) \times \min_{p' \in t} \{m(p', \tau)\} - \sum_{t \in \bar{*, p}} \lambda(t) \times \min_{p' \in t} \{m(p', \tau)\}.
\]

The equation holds since \( t \in En(p) \) is equivalent to \( t \in \bar{*, p} \), \( t \in Ex(p) \) is equivalent to \( t \in *, p \) and \( p' \in Ex(t) \) is equivalent to \( p' \in \bar{*, t} \). Hence the ODEs generated by a PEPA model are the same as those for the associated CPN under infinite server semantics.

Figure 4 shows the CPN obtained from the activity graph for the servers example, together with its initial marking. \( Pre \) and \( Post \) can easily be determined, and it is clear, for example, that \( Ex(C) \) is the same as the post set of the place with initial marking 100.

**FROM A CPN TO A PEPA MODEL**

It is more complex to do the reverse translation. First, it is necessary to add implicit places to the net and to show this does not affect the ODEs that are obtained. Once implicit places have been added, the net can be transformed to a PEPA model using an existing algorithm (Hillston et al. 2001). An implicit place is a place in a net that can be removed without changing the behaviour of the net (Silva et al. 1998).

**Definition 5** An implicit place \( p \in P \) is a place such that whenever \( t \in p^* \) is enabled at time \( \tau \), \( m(p, \tau) \geq \min_{p' \in t} \{m(p', \tau)\} \).

The addition of an implicit or complementary place for each place in a net is called complementation. For each place \( p \), a new place \( \bar{p} \) is added such that \( *\bar{p} = p^* \) and \( p^* = *\bar{p} \) using the following construction (Hillston et al. 2001). The construction requires that the net be bounded, and for each place \( p \) there is an upper bound \( b(p) \) on the value that the place \( p \) can have in any marking reachable from \( m_0 \).

**Definition 6** Given a net \( S = (N, m_0) \) with \( N = (P, T, Pre, Post, \lambda) \), its complementation is a net \( S' = (N', m'_0) \) with \( N' = (P', T, Pre', Post', \lambda) \) where \( P' = P \cup \bar{P} \) with \( \bar{P} = \{p | p \in P\} \).

\[
Pre' = \begin{bmatrix} Pre \\ Post \end{bmatrix} \quad Post' = \begin{bmatrix} Post \\ Pre \end{bmatrix} \quad m'_0 = \begin{bmatrix} m_0 \\ b - m_0 \end{bmatrix}
\]

A new place \( \bar{p} \) can be shown to be implicit by proof by contradiction on the bound \( b(p) \). Also \( m_R(p, \tau) = m_S(p, \tau) \) for all \( p \in P \), where \( m_R \) refers to the marking in the context.
of the net $R$. Furthermore $enab_{S'}(t, \tau) = enab_{S}(t, \tau)$ for all $t \in T$ where $enab_{S'}$ refers to the enabling degree in the context of the net $R$. The following ODEs can be obtained from the CPN $S'$ created by complementation of $S$.

$$\frac{dm_{S'}(p, \tau)}{d\tau} = \sum_{t \in P_r} \lambda(t) \cdot \min_{p' \in t} \{m_{S}(p', \tau)\}$$

$$- \sum_{t \in P^*} \lambda(t) \cdot \min_{p' \in t} \{m_{S}(p', \tau)\}.$$

The focus here is on the original places in $P$. Note that

$$\min_{p' \in t} \{m_{S'}(p', \tau)\} = \min_{p' \in t \cap P} \min \{m_{S}(p, \tau), m_{S}(p', \tau)\},$$

since for each $p$, $m(p, \tau) \geq \min_{p' \in t \cap P} \{m(p', \tau)\}$, and so the value of the marking at that implicit place can be ignored when determining the minimum. Considering a place $p$ in $P$, we obtain the same equation as that obtained before complementation.

$$\frac{dm_{S'}(p, \tau)}{d\tau} = \sum_{t \in P_r} \lambda(t) \cdot \min_{p' \in t \cap P} \{m_{S}(p', \tau)\}$$

$$- \sum_{t \in P^*} \lambda(t) \cdot \min_{p' \in t \cap P} \{m_{S}(p', \tau)\}.$$

The net formed by complementation can be translated into a PEPA model using the algorithm presented by Hillston et al. (Hillston et al. 2001). The algorithm has one minor change; the rates are not passive in the sequential components. This is necessary to meet the restriction on the form a PEPA model can take and does not affect the correctness of the algorithm. It is possible to derive ODEs from this PEPA model as done above. However, it is simpler to use the high/low approach for modelling biological systems in a reagent-centric fashion (Calder et al. 2005). In this approach, each reagent can be at a high or low concentration, and this is modelled using a single component with two states. The ODEs are then obtained for the reagent rather than for each of its states. In the cited reference, mass action is used in the ODEs since it is appropriate for the biological model. In the context of this paper, $\min$ is the appropriate function to use as it captures possible synchronisations.

The PEPA model obtained by complementation can be viewed similarly, so component $C_p$ is the high concentration and its derivative $C_p'$ is the low concentration. An activity graph and an activity matrix can be constructed (Calder et al. 2005).

**Definition 7** An HL activity graph is a bipartite graph $(N, A)$. The nodes are partitioned into $N_t$ the activities and $N_p = \{C_p \mid p \in P\}$. $A = (N_t \times N_p) \cup (N_p \times N_t)$, where $a = (n_t, n_p) \in A$ if $n_t$ is an entry activity of $C_p$ (and an exit activity of $C_{p'}$) and $a = (n_p, n_t) \in A$ if $n_t$ is an exit activity of $C_p$ (and an entry activity of $C_{p'}$). The HL activity matrix $M_{a}$ is an $N_p \times N_t$ matrix and entries are defined as follows.

$$M_a(C_p, t) = \begin{cases} +1 & \text{if } t \text{ is an entry activity of } C_p \\ -1 & \text{if } t \text{ is an exit activity of } C_p \\ 0 & \text{otherwise.} \end{cases}$$

Note that here the activities are just the transitions $T$ from the net. However, $N_p$ is the same size as the set of original places, and hence the activity graph is not isomorphic to the complemented net but rather to the original CPN. The ODEs that can be extracted from this model are as follows.

$$\frac{dN(C_p, \tau)}{d\tau} = \sum_{(t, \lambda(t)) \in En(C_p)} \lambda(t) \times \min_{C_{p'} \in Ex(t, \lambda(t))} \{N(C_{p'}, \tau)\}$$

$$- \sum_{(t, \lambda(t)) \in Ex(C_p)} \lambda(t) \times \min_{C_{p'} \in Ex(t, \lambda(t))} \{N(C_{p'}, \tau)\}.$$
Figure 6: A CPN with Complementation

<table>
<thead>
<tr>
<th></th>
<th>prepare</th>
<th>serve</th>
<th>finish</th>
<th>reset</th>
<th>fail</th>
<th>repair</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{p1}$</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_{p2}$</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_{p3}$</td>
<td>0</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_{p4}$</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_{p5}$</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$C_{p0}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\frac{dm(p_1, \tau)}{d\tau} = f.m(p_3, \tau) - p.m(p_1, \tau)
\]
\[
\frac{dm(p_2, \tau)}{d\tau} = p.m(p_1, \tau) - s.min\{m(p_2, \tau), m(p_4, \tau)\}
\]
\[
\frac{dm(p_3, \tau)}{d\tau} = s.min\{m(p_2, \tau), m(p_4, \tau)\} - f.m(p_3, \tau)
\]
\[
\frac{dm(p_4, \tau)}{d\tau} = r.m(p_5, \tau) + e.m(p_6, \tau) - s.min\{m(p_2, \tau), m(p_4, \tau)\}
\]
\[
\frac{dm(p_5, \tau)}{d\tau} = s.min\{m(p_2, \tau), m(p_4, \tau)\} - r.m(p_5, \tau) - a.m(p_5, \tau)
\]
\[
\frac{dm(p_6, \tau)}{d\tau} = a.m(p_5, \tau) - e.m(p_6, \tau)
\]

Figure 7: The Activity Matrix Obtained from the Unreliable Servers PEPA Model and the ODEs Obtained from the Unreliable Servers CPN in Figure 1

Figure 6 shows the unreliable servers example from Figure 1 under complementation with appropriate bounds. The ODEs from the uncomplemented net are given in Figure 7. From the complemented net the PEPA model in Figure 8 is obtained. The HL activity matrix is given in Figure 7, and the same ODEs as in Figure 7 can be extracted from this matrix.

SERVER SEMANTICS

As mentioned above, there are two types of server semantics. In infinite server semantics, a transition can fire simultaneously as much as it is enabled. This is also called marking dependent or variable speed (David and Alla 2005). For the finite server or constant speed case (David and Alla 2005), the number of simultaneous firings of transition is bounded. The transformations above have shown that the ODE semantics of PEPA are infinite server semantics. Infinite server semantics are the more general case. In the discrete case, finite server semantics can be obtained in a net with infinite server semantics by modifying the net. This requires the addition of a place with $k$ tokens with arcs to and from the transition which is to be bounded (Mahulea et al. 2006).

A similar approach can be taken in PEPA. When a particular activity is to be limited, a new component can be added of the form $B_{\alpha} \overset{\text{def}}{=} (\alpha, r).B_{\alpha}$ with $k_{\alpha}$ copies where $(\alpha, r)$ is the activity to be constrained. Letting $\mathcal{N}(C)$ refer only to the original components and their derivatives, then the resulting ODEs can be expressed as

\[
\frac{d\mathcal{N}(D, \tau)}{d\tau} = \sum_{(\alpha, r) \in E_X(D)} r \times \min\{k_{\alpha}, \min_{C \in E_X(\alpha, r)} \{N(C, \tau)\}\}
\]

Note that the ODE each $B_{\alpha}$ has the form $\frac{dN(B_{\alpha})}{d\tau} = 0$ since the number of copies is always $k_{\alpha}$. An example is given in Figures 10 and 9. This represents senders that collect data and pass it on to receivers to deliver the data. The more abstract version (on the left in Figure 10 and SendRec in Figure 9) has infinite server semantics and represents the system without constraints. The second version (on the right and SendRecK) has additional places/components that constrain the transitions/activities. This represents a more concrete version of the system where there is a medium between senders and receivers that only permits a certain number of exchanges at one time, and both the sender and receiver must each interact with a buffer of limited capacity in collecting or delivering the data. The behaviour over time of this system
\[ C_{p_1} \triangleq (\text{prepare}, p).C_{p_1} \]
\[ C_{p_2} \triangleq (\text{serve}, s).C_{p_2} \]
\[ C_{p_3} \triangleq (\text{finish}, f).C_{p_3} \]
\[ C_{p_4} \triangleq (\text{prepare}, p).C_{p_2} \]
\[ C_{p_5} \triangleq (\text{serve}, s).C_{p_5} \]
\[ C_{p_6} \triangleq (\text{repair}, e).C_{p_6} \]
\[ C_{p_7} \triangleq (\text{fail}, a).C_{p_7} \]

\[ M \triangleq C_{p_1}[200] \times \{ \text{prepare}, \text{finish} \} \times \{ \text{serve} \} \times C_{p_2}[200] \times \{ \text{serve} \} \times C_{p_3}[100] \times \{ \text{serve}, \text{reset}, \text{repair} \} \times C_{p_4}[100] \times \{ \text{serve} \} \times C_{p_5}[100] \times \{ \text{fail} \} \times C_{p_6}[100] \]

Figure 8: Unreliable Servers Example

\[ P_1 \triangleq (t_2, \lambda_2).P_2 \]
\[ P_2 \triangleq (t_1, \lambda_1).P_1 \]
\[ P_3 \triangleq (t_2, \lambda_2).P_4 \]
\[ P_4 \triangleq (t_3, \lambda_3).P_3 \]
\[ B_{t_i} \triangleq (t_i, k_{t_i}).B_{t_i} \]

\[ \text{SendRec} \triangleq P_1[100] \times \{ \text{serve} \} \times P_3[100] \]
\[ \text{SendRecK} \triangleq (P_1[100] \times \{ \text{serve} \} \times B_{t_1}[k_{t_1}]) \oplus (B_{t_2}[k_{t_2}] \times \{ \text{serve} \} \times P_3[100] \times \{ \text{serve} \} \times B_{t_3}[k_{t_3}]) \]

Figure 9: PEPA Models of Sender and Receiver Example

\[ s(t, \tau) = \begin{cases} \lambda'(t), & \text{if } \forall p \in \ast t, m(p, \tau) > 0 \\ \min\{\lambda'(t), \min_{p \in \ast t} \left\{ \sum_{t' \in \ast p} \frac{f(t', \tau) \cdot \text{Post}(p, t')}{\text{Pre}(p, t)} \right\} \}, & \text{otherwise.} \end{cases} \]

In the case that enab(t, \tau) > 0 then the flow is just the firing speed of the transition. Otherwise, the flow through the transition is determined by the minimum of the speeds of the transitions supplying the empty places that supply the transition. Determining the speeds requires the solution of a linear programming problem (David and Alla 2005). Hence for ordinary nets, the ODEs have the form

\[ \frac{dm(p, \tau)}{d\tau} = \sum_{t \in \ast p} f(t, \tau) - \sum_{t \in \ast p} f(t, \tau) \]

with \( f(t, \tau) \) defined as above, and with the \( \text{Pre} \) and \( \text{Post} \) terms equal to one. These equations result in linear piecewise ODEs (Mahulea et al. 2006). The net behaves as described by a set of equations until a place empties, and then the speeds change since the enabling of the net has changed (a place is now empty). This switching between equations continues until a steady state is reached where no more changes in the marking can happen. This steady state is guaranteed if there is no conflict in the net (David and Alla 2005).

Finite server semantics also result in linear piecewise ODEs. Changes between equations occur because of the minimum function. Hence the ODEs for both types of server semantics can be viewed as a hybrid system, i.e., a system with both discrete and continuous behaviours. The continuous behaviour in this case are the sets of ODEs and the discrete behaviour is the switching from one set of ODEs to another due to a specific event, either the change in the minimum or the emptying of a place. Hybrid systems are an area of ongoing research.

can then be expressed using the approximation given by the ODEs in (\ast).

However, Silva and Recalde (Silva and Recalde 2005) argue that in the continuous case, infinite and finite server semantics are about two different relaxations of the model for approximation; many servers and many clients for infinite, and few servers and many clients for finite server semantics and are therefore different. In their view of finite server semantics, a transition has a maximal firing speed \( \lambda'(t) \) at which it can perform representing \( k \) times the speed of a single server (Mahulea et al. 2006) and a different expression is required for the flow of a transition.
A question of interest is how these the finite server ODEs compare to the explicitly constrained infinite server ODEs. Transforming the ODEs in (§) to the equivalent Petri net description gives

\[
\frac{dm(p, \tau)}{d\tau} = \sum_{t \in \pi} \lambda(t) \cdot \min\{k(t), \min_{p \in \pi^t} \{m(p, \tau)\}\} - \sum_{t \in \pi} \lambda(t) \cdot \min\{k(t), \min_{p \in \pi^t} \{m(p, \tau)\}\}
\]

where \(k(t)\) is the equivalent of \(k_\alpha\). Since \(\lambda(t)\) represents the rate at which transition \(t\) can fire, and \(\lambda'(t)\) represents the speed at which \(k(t)\) servers can be served, it can be assumed that \(\lambda(t), k(t) = \lambda'(t)\). Hence the question becomes a comparison of the terms

\[
\phi(t, \tau) = \lambda(t) \cdot \min\{k(t), \min_{p \in \pi^t} \{m(p, \tau)\}\}
\]

and

\[
f(t, \tau) = \begin{cases} 
\lambda(t), k(t), & \text{if } \forall p \in \pi^t, m(p, \tau) > 0 \\
\min\left\{\lambda(t), k(t), \min_{p \in \pi^t} \left\{\sum_{t \in \pi^t} f(t', \tau)\right\}\right\}, & \text{otherwise.}
\end{cases}
\]

Clearly the closer \(k(t)\) is to zero, the closer the values of \(\phi\) and \(f\) will be since the term involving \(k(t)\) is likely to be the minimum and hence both equation are likely to have the value \(\lambda(t), k(t)\). Understanding in more detail the relationship between these three approximations is future research.

**CONCLUSION**

This paper has shown how to construct an ordinary timed continuous Petri net from a PEPA model (using a subset of the language) and vice versa. Moreover it has been shown that when approximated continuously, the behaviour of both can be characterised by the same set of coupled ODEs. Furthermore the continuous approximation using PEPA has infinite server semantics. This paper has established links between two continuous approaches to modelling the performance of systems, one graphically based and the other textual, and hence techniques and results for one approach can be applied to the other. Both continuous approximation approaches are suitable for systems with many identical components.

Questions for ongoing research include how robust these modelling approaches are when component numbers decrease, and how the appropriate ODEs can be obtained when modelling large numbers of clients and a much smaller number of servers.

**ACKNOWLEDGEMENTS**

The author thanks Jane Hillston for her comments. The author is supported by the EPSRC SIGNAL Project, Grant EP/E031439/1. Part of this research was conducted while the author was on sabbatical leave from the University of the Witwatersrand.

**REFERENCES**


**BIOGRAPHY**

VASHTI GALPIN completed her tertiary education up to MSc level at the University of the Witwatersrand, Johannesburg, South Africa. She was awarded her PhD in Computer Science by the University of Edinburgh in 1998. Her thesis considered the metatheory of process algebras. After completing a one-year postdoctoral fellowship at the University of the Witwatersrand, she worked there as a lecturer for seven years. During this period her research focus was women in computing and computer science education research. She is currently a research associate in the Laboratory for Foundations of Computer Science in the School of Informatics at the University of Edinburgh and conducts research into stochastic process algebras as well as their application to modelling in systems biology.
A PERFORMANCE DRIVEN MODELLING APPROACH FOR SOA BASED APPLICATION

Mauro Iacono  
Dipartimento degli Studi Europei e Mediterranei  
Seconda Università degli Studi di Napoli  
Belvedere Reale di San Leucio,  
81100 Caserta, Italy  
mauro.iacono@unina2.it

Stefano Marrone  
Dipartimento di Matematica  
Seconda Università degli Studi di Napoli  
Via Vivaldi 43, 81100 Caserta, Italy  
stefano.marrone@unina2.it

KEYWORDS  
Workflow performance, Service Oriented Architecture, Models Composition, Petri Nets, Non-functional requirements.

ABSTRACT

Service Oriented Architectures are assuming a more and more relevant importance in integrating distributed systems by means of atomic services, allowing a coarse-grain reuse of existing solutions. Compositional systematic modeling approaches can improve the quality of applications built with this logic: many open issues still drive research on this topic, including the need for methodologies enabling the assessment of both functional requirements (such as correctness) and non functional requirements (such as performances and reliability) in the early phases of design. Since literature deeply analyzes the first kind of requirements, in this paper we try and address the second kind, and introduce a modeling framework that supports the of performance analysis of coarse-grain SOA based applications in the early phases of the design process. This is made by exploiting formal methods (in particular by means of Generalized Stochastic Petri Nets) to anticipate performance requirements compliance verification when possible, given a basic statistical characterization of the services to be integrated.

INTRODUCTION

The need for a scalable and cost-containing approach to business development and services delivery is a compelling urgency for medium and small enterprises in order to stay on the market and be competitive. This need is specially evident when creating and managing temporary organizations, born for a single project and composed by independent units that are involved in more different organizations. VO (and Virtual Teams, that implement the different functions in VO) need flexible, distributed and interoperable information systems architectures in order to lower the overall costs of operations and to seamlessly reuse existing components. Service Oriented Architectures (SOA) promise to be a consistent solution for this kind of problems (Camarinha-Matos 2002). The SOA philosophy considers delivery of software as a service the key approach to software reuse in the next years. A service is a self-consistent piece of software that offers an open interface to its functions towards users: although the idea is not revolutionary nor so new in itself, the novel issue that constitutes the real importance of this approach are the availability of mature standards and technologies to support the integration of the basic bricks of SOA (usually based on the Web Services (WS) technology) and the achievements of ongoing research on methods and technologies aimed to cope with problems regarding coordinated execution of services (orchestration and choreography, usually applied to WS). Together with coordination, other open issues are now part of current research in the field, such as services composition with a special attention to formalization for correctness and non-functional requirements verification.

In the most common scenario, applications compose geographically distributed coarse grain services belonging to different organizations. We believe that as correctness of this composition is needed to build an application, the fulfilment of non-functional requirements determine the probability of success for the application in a competing global market. This implies the need for non-functional early phases verification of the application.

Building applications by services composition implies delegating a wide part of the business code to existing services. One of the hardest task implied by this activity is the evaluation of service features in order to choose the one that best fits the applications purposes: due to inaccessibility of the services inner implementations, the only possible approach is to consider all services as black boxes about which only some observable characterization is useful. According to what previously said, this criterion must be applied to both functional and non functional characteristics. This activity seems to be the most effective in order to compare different alternative integration architectures of services, even from a quantitative point of view.

In this paper we present a basic modelling framework for performance assessment of coarse grain, generically distributed SOA applications in the early phases of development, by exploiting the modelling power of Petri Nets (in particular their stochastic extension Generalized Stochastic Petri Nets). Generalized Stochastic Petri Nets (GSPN) best fit this purpose because of their user friendliness and the possibility to provide quantitative performance evaluations.
The paper is organized as follows: after this introduction, a short related works section will introduce the principal contributions in the field. A section illustrating the modelling framework and an example of application of the methodology will lead to conclusions.

**RELATED WORKS**

SOA is a widespread architectural approach for building software applications that use services available in a network, such as the Internet, and it represents the widest accepted model to design geographical distributed systems. It promotes loose coupling between software components, improving their reusability (Erl 2005). On the other hand this paradigm had already proved the high added value in a global market populated by Virtual Organizations (VOs) (Camarinha-Matos 2002): these entities are best defined as temporary aggregations of heterogeneous companies under a well-specified objective or mission (Camarinha-Matos and Afsarmanesh 1999).

The problem of the evaluation of SOA-based solutions has also been studied in scientific literature, that is rich of such papers, the most of which deal with the verification of the correctness of the resulting application. This characteristic has been exhaustively studied by means of different kinds of formal methods (pi calculus, logic, model checking).

Another stream is related to the studies of workflows by means of Petri Nets: several works have been produced on this topic and all of them deal with the verification of the correctness of the workflow. A detailed description of this problem is in (van der Aalst 1998) that surveys this argument: moreover other contributions on this research come from (van der Aalst 2002; van der Aalst 2000; Kob et al. 2005). Moreover, we must cite the transformational approach presented in (Hintz et al 2005), related to the development of automatic transformation between BPEL and Petri Nets.

Other studies involve the use of formal methods in the verification process of non-functional properties such as security (Bhargavan et al. 2006; Kim and Biswas 2006). Services composition performance has been studied in different ways: in (Hamadi and Benatallah 2003; Ho et al. 1998). Web Services performance are studied by mean of direct measuring and statistical techniques while other approaches involve use of formal methods. These latter studies use Performance Evaluation Process Algebra (PEPA) as in (Gilmore and Tribastone 2006) or timedCCS Process Algebra as in (Martineilli and Matteucci 2007); other approaches rely on the applications of Timed Automata in (Diaz et al. 2006): performance of Web Services have been also studied by means of Petri Nets in (Deng et al 2007; Hamadi and Benatallah 2003 ). The most similar approach is in (Xia and Chang 2006) where a measurement and evaluation architecture is presented.

The main original contribution of this paper is a compositional design methodology by which a performance model of a SOA-based application can be built by composing alternative performance reusable submodels and the main business process, just as real (reusable) services build up the application itself: inexpensive performance evaluations on alternative compositions of submodels guide design choices. At the best of our knowledge, our contribution is the only lightweight approach for designing performance-oriented geographically distributed SOA applications.

Composition of Petri Nets models are in the scope of our research. Literature offers several approaches (Sibertin-Blanc 1994; Lakos 1995) one of the most significant is the Cooperative and Communicative Nets (Sibertin-Blanc 1994) whose application in computer and communication systems have been also explored (Franceschini et al. 2003; Franceschini et al. 2004).

**METHODOLOGY**

We propose a methodology based on the analysis of both the “in the large” part of the system and existing services with GPSPN. The two elements of the analysis have different characteristics and in a first step should be evaluated separately.

An important prerequisite of the analysis is the systematic collection of all available services and related black box performance characterization. This collection process should categorize available services into groups of functionally equivalent services tagged with related information. The groups, named after the general definition of the functionality offered by processes in each group, form the services catalogue, that is used to identify the atomic granularity level to be considered by the subsequent “in the large” design step. If the analysis of services is executed by considering services offered by different providers on a per-provider basis, different catalogues are obtained containing the groups. In this (more general) case, names of the groups are annotated with a proper semantic description to enable the subsequent unification step.

Once all local per-provider services catalogues are available, they are unified, that is integrated into a single catalogue, by exploiting the semantic annotation in order to avoid groups duplications due to different names. The result catalogue is the global services catalogue, that in the rest of this section is referred as services catalogue tout court.

Functional specifications of the application can thus be developed and detailed with reference to the services catalogue. Whenever a functionality is not available in the service catalogue, another external (outsourced, acquired) or new (“in-house”, to be developed “in the small” according with one of the known design cycles) service must be provided while taking into account additional costs in the project. Obviously, in some cases the fact that a service is not available (in the catalogue or on the market) can signal the need for a further investigation on the possibility of fragmenting complex functionalities into smaller, services-covered atomic functionalities by decomposition of specifications. New services covering missing functionalities are added to the services catalogue in order to obtain and keep a reusable corpus of knowledge about reusable resources.

Once all functionalities are covered, the specifications are detailed as several alternative versions of a business process (BP) using services (eventually specified as a BPEL document) or, equivalently, a workflow. In the rest of the paper we will indifferently consider BPs or workflows,
without affecting the generality of the contents. In case of complex BPs, non-critical paths can be neglected in the analysis, thus reducing its complexity to paths that are classified as critical for the frequency with which they will be enabled of their length in steps or in total completion time. These considerations on complexity can be obtained by a proper analysis of the specifications themselves. Starting from this simplified version of the BP, service invocations represent the interfaces towards services submodels, and will be completed according to the single choreography. After this step, it is necessary to translate the BP into a formalism that can properly cope with performance evaluation as Generalized Stochastic Petri Nets. This process is not described here but several approaches to such translation can be find in literature and are cited in the next Section.

For each service, choreography (eventually defined as a WS-CDL document) and available statistical performance measures (depending on the service choreography itself, total or partial service completion or response times) should be available in order to classify and build the service model. Performance measurement techniques are not in the scope of this paper, so we will assume that a proper characterization has been obtained. By using a sender-receiver two-sides choreography description of the normal invocation(s) of each service, it is possible to identify all and only the interactions needed for a proper integration of the service in the application. Interactions define the interfaces of the GSPN submodel encapsulating the service behaviour while performance annotations allow the correct instantiation of needed timed transitions. It is possible to obtain the GSPN submodel as shown in the next Section.

An intermediate communication GSPN submodel will be generated to interface the service subnet and the BP subnet by using again the choreography information to extract the communication protocol. The creation of the intermediate net is described in the next Section. For each alternative combination that has to be considered, subnets obtained during the process are fused into a single GSPN that is evaluated. The modular design of the model allows subnets to be reused during the process.

**MODEL CONSTRUCTION**

In order to show how to obtain the performance model, in this section proper guidelines will be given related to each of the three kinds of submodels: BP, services and interconnections.

**Modelling the BP**

As previously stated, scientific literature widely faced the problem of the translation of a generic BP or workflow into a Petri Net. A SOA based application, that uses basic services by means of a complex orchestration, can be seen as a BP or a workflow.

Classical workflow analysis highlights several workflow operators focused on the control and data flow inside the workflow itself. Language construct as loop, deterministic or stochastic choice, fork and join can be easily modelled by Petri Nets (Deng et al. 2007; Hamadi and Benatallah 2003): anyway, these approaches do not take into consideration two important aspects of the problem that are relevant for the goals of this paper: the kind of service invocation patterns and the communication mechanisms.

About the first aspect, four main possible cases of services invocation patterns can be defined, by combining the two main invocation methods (asynchronous and synchronous) with the two return behaviours of services (with or without return value), as shown in Table 1. Out of them, three main acceptable invocation patterns can be defined, analysed and translated into GSPN.

<table>
<thead>
<tr>
<th></th>
<th>With return</th>
<th>Without return</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synchronous</strong></td>
<td>blocking</td>
<td>not applicable</td>
</tr>
<tr>
<td><strong>Asynchronous</strong></td>
<td>non blocking</td>
<td>simple message</td>
</tr>
</tbody>
</table>

Table 1: Service invocation patterns

Starting from Communicative Nets (Sibertin-Blanc 1994), in a previous work the authors defined a methodology for SWN composition based on invocation patterns both for sender and for receiver side (Franceschinis et al. 2003; Franceschinis et al. 2004), whose application in computer and communication systems have been also explored. By properly applying the main ideas of these works to the previously proposed classification, it is possible to translate into GSPN fragments the three invocation pattern models. Results are depicted in Figure 1, Figure 2 and Figure 3.

**Figure 1 : Blocking Service Invocation Pattern**

**Figure 2: Non blocking invocation pattern**

**Figure 3: Simple message invocation pattern**

The second aspect (communication mechanism) is related to the interconnection between the BP and services and will so be considered in the third subsection.

**Modelling services**

Modelling and characterization of web services from the user point of view is a topic still not very explored by
literature. In this paper we suggest a black box approach that allows us to identify a statistical model of response times by simple measurements. Such an approach is based on these performance data in order to use a simple GSPN model that does not take into account service implementation and access details.

![Deterministic timing service model](image)

**Figure 4: Deterministic timing service model**

![Stochastic timing service model](image)

**Figure 5: Stochastic timing service model**

![Congested service model](image)

**Figure 6: Congested service model**

For sake of simplicity we will focus our attention on the services with a return value: a similar reasoning can be done for the others. As already applied for service invocation patterns, basic services can be easily classified. A first tentative of such classification finds three cases: a stochastic timing service implementation, a deterministic timing service and the congested one. These models are depicted in Figure 4, Figure 5 and Figure 6.

The basic structure of a service GSPN model is obtained by a place in which request are accepted, a body that elaborate the requests (consuming time according to the characterization) and a transition that sends back a return message. Obviously, more complex statistical characterization can be obtained as in Figure 6 (two periodical alternating behaviours due to different workloads) and services with more interfaces can be modelled with multiple instances of these simple submodels.

**Modelling the interconnection**

Due to the need to choose Internet as underlying communication media, services performance and reliability suffer from the “best effort” behaviour of the Internet: according to this statement simple communication models are not sufficient to cope with this problem. In our methodology the modelling of unreliable or congested network is an activity that is taken into account by black-box modelling of basic services shifting the network delay into the service average rate parameter, so, for the scope of this paper, we will consider only the case of “instantaneous” communication that can bee seen in Figure 7.

This network can be properly used to fuse senders and receivers according to what we have proposed in the two previous subsections. The underlying mechanism is constituted by place and/or transition fusion of a GSPN that is widespread used in several methodologies and supported by existing tools as Algebra (Bernardi et al. 2001). Figure 8 and Figure 9 show how this mechanism works in case of blocking service invocation patterns and simple stochastic server. Boxes in Figure 8 show the place and the transition that have to be fused.

![Instantaneous communication pattern](image)

**Figure 7: Instantaneous communication pattern**

![Joining network](image)

**Figure 8: Joining network**

![Result network](image)

**Figure 9: Result network**

In order to prevent the need for rewriting the workflow net when trying different alternative services sets, the interconnection net includes all the details for accessing the service on one interface, while offers a simple invocation pattern as the one shown towards the workflow. Thus the interconnection net describes the logic needed to complete the communication protocol described by the service choreography. A simple synchronous invocation of a service with a single interface for receiving a request and sending a reply will be modelled as a simple net made of one transition and one net per interface, while the invocation of a more complex service, for which a complete invocation is made of several request/reply interface interactions will include a state logic, that has to be designed case by case.
MODELLING EXAMPLE

To give an example of application of the approach, in this Section a simple single implementation of a single functionality of an application will be presented. The focus is on a VO that performs maintenance of industrial plants. This VO has four partners (A, B, C, D). The functionality is the handling of a customer request that can result into delivery of supplies or the intervention of a maintenance team. The application needs the coordination of a number of facilities that are shown in Table 2, together with a mapping of services available at partners servers.

As from Table 2, all needs are covered by classes on the service catalogue with the exception of EC, that will be acquired from a third party. For the sake of space, only one BP GSPN will be showed in Figure 10, in which, besides service invocation interfaces, it is worth to notice the timed transition describing the effects of local computations and a cycle describing repeated invocations of a service.

Table 2: Functionalities and mapping

<table>
<thead>
<tr>
<th>Description</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance Teams</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Management (TM)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduling (SC)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Communications (IC)</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Communications (EC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e-Procurement Order Placement (EP)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shipping Management (SM)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

To show a GSPN modelling a service, it is possible to consider the EP service which choreography is in Figure 11. This service has two input interfaces (Lookup and Buy) with related replies, and the performance model of Lookup consists in two alternated behaviours (approximated as two different exponentially distributed stochastic behaviours which alternation is an exponentially distributed behaviour itself) due to non homogeneous workload. The service GSPN is in Figure 12, boxes indicate fusion points with the other nets.

To interconnect the service and the BP, according to Figure 11, the GSPN must capture the repeated invocation of Lookup (until success, described as a probabilistic choice). The interconnection GSPN is depicted in Figure 13.
CONCLUSIONS AND FUTURE WORKS

The paper proposed an approach to the evaluation of non functional characteristics of SOA based applications for the support of VO. This is addressed by means of Generalized Stochastic Petri Nets for performance evaluations and net composition. This approach can be considered just the first step and the first consistent result of a promising research activity.

Besides a further development and formalization of this approach, planned future works include the extension towards a complete methodology oriented to different non functional characteristics (reliability, performability) by adopting multiple formalisms, the introduction of more sophisticated PN formalism as Stochastic Well-formed Nets and the parallel development of a proper methodology for the black box evaluation of services characteristics.

REFERENCES


EMERGENCY RISK MANAGEMENT
LARGE SCALE SYSTEMS
Revealing the impact of information provision on disaster relief operations by means of agent-based modeling and simulation

Gerald Lichtenegger
Siegfried Voessner
Department of Engineering and Business Informatics
Graz University of Technology
Kopernikusgasse 24/III
A-8010 Graz, Austria
E-mail: {gerald.lichtenegger|voessner}@tugraz.at

KEYWORDS
Disaster management, disaster relief, agent-based modeling, agent-based simulation, information management;

ABSTRACT
Today technological progress enables more and more advanced concepts and solutions in the field of disaster management. However, in disaster relief this progress cannot be turned into results to the extent it is supposed to be. Among others the allocation of and the dealing with (battlefield) information has turned out to be one major weakness causing troubles in the course of disaster relief operations (DROps).

In this paper a generic, agent-based model of a disaster relief network is presented. The primary objects of investigation, unit deployment and mission execution, are approached considering the commonalities as well as taking into account the differences of the participants in disaster relief operations. By varying allocation and content of the information in the disaster relief network, changes in the characteristics of unit deployment and mission execution can be observed. This approach supports both, narrowing down the application areas of information and communication technology (ICT) and pointing out the effects caused by changes in organizational structures.

INTRODUCTION
In the recent past societies all over the world have had to suffer more often from different types of disasters – natural and human-made ones. Regarding the likeliness of disaster occurrence the future prospects are even worse. Focusing on the segment of disaster relief the issues information allocation and information content are, among others, of special interest. The enhancements in ICT support almost every desired scenario. However, especially the lack of information at the right point and at the right time as well as the provision of incomplete or improper content results in time delay with unit deployment and so causes severe trouble in mission execution.

Disaster relief system

Looking at today’s prevailing national disaster relief systems on can say that all of them are quite similar. Depending on the different institutions that are involved in DROps, governmental and non-governmental ones, and what is stated in legal requirements, more or less all of these systems show a hierarchical three staged structure (cp. Figure 1).

![Figure 1: Basic structure of disaster relief systems](image)

At the lowest level, operation execution, all the activities and the field work performed directly at the disaster site and its vicinity are located. This covers for example the missions of medical services, fire fighters, police and armed forces.

One stage above, at the level of operations planning and control, the tasks dealing with unit deployment and replenishment as well as the controlling of the different DROps running simultaneously at the basis are situated. At operational headquarters the decisions concerning the allocation of disaster relief units for a specific operational area are taken. Furthermore, battlefield information is collected, aggregated and re-provided for further use. This is performed by cross-functional groups staffed with people from civil authorities, governmental and non-governmental organizations and other specialists.

At the top level, strategy and support, all activities in the course of the disaster relief process are monitored based on the information obtained from the levels below. At this stage the “big picture”, the disaster overview is generated. This is the place where an overall strategy for disaster relief is developed (von Kirchbach et al. 2002). Moreover, with the ability to overlook, the provision of additional resources, units, capacity etc. that might be requested is a major
business. At this level the responsibility is solely located at the (federal) authorities (Harrald 2006).

Flow of information

Due to this widespread structure of disaster relief systems in combination with an almost uniform understanding of command and control, the flow of information follows the channels set up and used by the “chain of command” (cp. Figure 2).

![Figure 2: Directions of information flow](image)

Unlike the intended vertical flow of information there are hardly any rules for distributing and exchanging information at the same level. The advantage of transferring a well known and proven structure faces some additional disadvantages which, under certain circumstances, decrease the performance of the whole system. First of all, neither the information flow within the system is automated and independent from human-triggered interactions nor exists a standardized structure classifying the content provided that enables automated processing. Secondly, when losing the connection to higher-level respectively commanding units or headquarters, an alternative exchange of high-priority battlefield information is hardly supported by the “chain of command”-structure (Haddow and Bullock 2005). Remembering major disasters across Europe and the United States this disadvantages had contributed to the following weaknesses in or problems with the execution of disaster relief operations (von Kirchbach et al. 2002; Townsend 2006; Harrald 2006):

1. No or inadequate prioritization of activities
2. Misdirection of field units and other resources
3. Forming of shadow organizations
4. By-pass communication and data exchange

In contrast to the obvious problems mostly caused by a lack of information at the level of operations planning and control (points 1 and 2) it is argueable that especially the occurrence of the problems mentioned in points 3 and 4 has not only negative effects in the course of disaster relief. When loosing the familiar ties of communication and information provision over a long period of time with no predefined backup available, the system (namely the units at the level of operation execution) tries to restabilize itself. This can be considered as a type of self-organization. In almost every case this is done by bringing together alternative resources in a temporary, localized organization and setting-up substitutions for the missing links in communication and data exchange. Under exceptional conditions this resilience prevents the disaster relief system from loosing its abilities of acting and reacting. These emergency procedures keep the system running. For several reasons most of the time that issues come up without an emergency situation (e.g. the loss of connections to commanding units). In fact, lacking an extensive (and uniform) amount of appropriate rules and predefined procedures in information exchange, many disaster relief systems support a behavior of its participants that might have negative impacts on DROps.

In the up-coming sections of this paper an ABM-approach that aims on answering the question to what extent information provision at the level of operation execution is still helpful is discussed. This is done by changing the “horizon of information” as well as the information content and by making use of the existing tendencies of self-organization mentioned above.

MODEL

A general model of a disaster relief system’s operations-level and the connection to its purpose of existence, the disaster itself, that has the potential to return verifiable results seems to be a “mission impossible”. Therefore, many research projects have set up very special models covering sub areas in the course of DROps.

The presented approach tries to face the challenge by focusing on a set of parameters that has substantial impact, especially in the launch phase of DROps: unit deployment and mission execution. Hence further factors that affect all of the chosen parameters the same way or causes changes in behavior only under very specific circumstances are left unaccounted.

Disaster

Very different approaches in classifying a disaster are used for certain purposes. In the presented model only some general classification parameters out of the following groups are of note:

- Incident (What)
- Location (Where)
- Impact (How much)

The group incident contains information regarding the different categories of work that has to be performed in the course of handling a special disaster. The categories have to be split-up so that they show as less overlap as possible. Also the categories have to be chosen in a way that the bulk of tasks performed by the field units can be assigned clearly.
Examples for categories of work are:
- Fire fighting
- Medical service
- Public safety
- Technical assistance
- Logistic
- ...

Closely related to the incident is the group impact. For each category of work the maximum workload that the smallest field unit is capable of is defined as a reference value. The disaster impact in a category $I_X$ is a multiple of this value. There are a few more related parameters to describe this group completely. For better understanding they are described in the section rules. One disaster of course has many categories whose impacts might be quite different. To locate the disaster site and to set it into relation to the field units it is necessary to name its location. This is done by assigning coordinates $(x,y)$ of a 2D simulation grid.

Field units

The field units (agents) are defined by:

- Category $X$
- Capability $c_X$
- Reaction time $t_R$
- Mission cycle $t_M$, $t_{off}$
- Life cycle $n_M$
- Location $x,y$
- Education
- Activity

Disaster relief units (field units) cover at least one category of work (main focus). However, almost every unit has expertise and therefore capacity in more than one category. The capability $c_X$ in one category might differ from the capability in another category. The capability of the reference unit (smallest unit applicable in a certain category) is the basis for calculating $c_X$. Depending on the unit size the capabilities are a multiple of the reference value. Reaction time $t_R$ frames the duration between alerting (requesting) the unit and its deployment. The two parameters representing the mission cycle give information about the maximum duration one unit is able to work non-stop on a mission ($t_M$) and the duration of the break that has to be taken for on-site regeneration ($t_{off}$). The life cycle $n_M$ defines the maximum number of mission cycles. When reaching that number the field unit has to return to its home base.

Home bases

Field units are located at home bases which can house several field units. All these units share the parameters assigned to its base. First of all this is the location on the map $(x,y)$. Furthermore the parameters education and activity are of note. Both of them are qualitative factors. Education describes the ability of one unit to interpret information concerning the demanded categories and capabilities in a proper way. The higher the value of education is the higher is the possibility of a proper reaction in unit deployment.

Activity deals with the unit’s tendency towards self-deployment with or without proper information. A low activity value represents only a minor chance that unit deployment will take place without an explicit request from higher-level or commanding units. Some authors call these two parameters the agents’ character (Takahashi 2006).

Rules

Performing a simulation on disaster relief activities requires some basic rules concerning mission execution:

1. *Mission execution is restricted to one category at a time* When deploying a certain unit to a mission one of the categories the unit is capable of has to be chosen. That choice is irreversible until the unit has ended its mission by completion or abandonment.

2. *Minimum capability $C_{min}$* Until the sum of capabilities in the desired category represented by the field units at the disaster site has not reached (or exceeded) the minimum capability the mission cannot start (cp. Figure 3).

$$\sum c_X$$

![Figure 3: Influence of capabilities](image)

3. *Maximum capability $C_{max}$* The maximum capability is the upper bound for the sum of capabilities allowed on site. Exceeding this value causes a decrease in mission progress (cp. Figure 4). It reduces the sum of capabilities provided by the units on site. This is covered by the factor $c_{red}$. The $C_{max}$ might be larger than the impact $I_X$. This means that exceeding the impact value doesn’t add additional power on mission execution.

4. *Impact has no time component* If the impact has no time component the relief is done by providing adequate capability. Thus the work on a category can be seen as completed when the sum of the capabilities provided by the field units $\sum c_X$ level the impact $I_X$. In this case the time component $t_X$ is set to the value of one time step in simulation.

5. *Impact requires lasting relief efforts* Unlike in point 4 in this case the completion of work can only be successful until the sum of the capabilities times their mission times, $\sum c_X * t_M$ reaches the value $I_X * t_X$ (cp. Figure 4).
Scenario

The total of information about the disaster, the field units that might get involved and the rules applied during the simulation process is called scenario. Some examples for parts of a scenario are shown in Tables 1 - 3.

<table>
<thead>
<tr>
<th>Location (x,y)</th>
<th>9</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fire fighting</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Medical service</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>Technical assistance</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Public safety</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Logistic</td>
<td>5</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2: Base

<table>
<thead>
<tr>
<th>Location (x,y)</th>
<th>EDUCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base number</td>
<td>3</td>
</tr>
<tr>
<td>Location (x,y)</td>
<td>15 4</td>
</tr>
<tr>
<td>Education</td>
<td>10</td>
</tr>
<tr>
<td>Activity level</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3: Unit (agent)

<table>
<thead>
<tr>
<th>Category</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home base number</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td></td>
</tr>
<tr>
<td>Fire fighting</td>
<td>1</td>
</tr>
<tr>
<td>Medical service</td>
<td>2</td>
</tr>
<tr>
<td>Technical assistance</td>
<td>3</td>
</tr>
<tr>
<td>Public safety</td>
<td>4</td>
</tr>
<tr>
<td>Logistic</td>
<td>5</td>
</tr>
</tbody>
</table>

SIMULATION

After having defined a scenario only a few settings are missing. At first the size of the initial operational area has to be defined. At the very beginning of the DROPs in this area the local and organizational involved bases and their units are located. It is assumed that in the course of the disaster relief process the number (capabilities) of field units located in that area is to low for a successful handling of the disaster so that “external” help is needed. Secondly, the procedures for sharing information beyond the borders of the initial operational area are to be assigned. The following procedures are available:

Expansion of the “horizon of information”

That means that certain information content is made available to bases, units etc. located outside the initial operational area. During the simulation process the expansion can be performed stepwise or at once till it reaches the maximum of the simulation grid.

Trim of information content

The information content provided in the course of expanding the horizon of information can differ: the notification that 1) capability is needed at a defined location without further information regarding incident or impact, that 2) a certain incident (categories are known) requires assistance with no information concerning the impact or that 3) a certain incident requires action to relief a given impact.

The combination of the chosen procedures applied on the scenario parameters will result in very different agent behavior. An example picture of a simulation grid is shown in Figure 5. Disasters that cover a greater area than represented by one raster square in the simulation grid have to be split up in several sub-disasters.

RESULTS

Since programming has not been finished yet no presentable simulation results are available at this point of time. Nevertheless some attention should be drawn to the desired results.

The questions that could be answered with the help of this simulation model differ regarding the point of view. As stated before the presented simulation model focus on the effects of information provision on unit deployment and mission execution at the level of operation execution. The questions to be answered are:

What settings in information provision lead to the fastest unit deployment?

How do the different settings of information impact affect unit deployment?
What settings in information provision lead to the most efficient mission execution (number of units on site, duration)?

How do the different settings of information impact affect mission execution?

Does a setting exist that provides improvements in unit deployment in combination with an efficient mission execution?

To answer this questions the completion of work at the disaster site (with respect to the time) with subject to the decisions made regarding to make battle field information available to not yet involved parties and the channels used (the choice for using different ICT and procedures will result in a provision of diverse content) has to be evaluated.

The expected results can give a valuable contribution to future investment decisions dealing with ICT. That will help to strengthen the channels of information provision that have the strongest impact on DROps. Furthermore the simulation results can help to point out what changes in the organizational structure of prevailing disaster relief systems should be made to enable progress in the areas the presented research is focusing on. Only the combination of organizational change and specific application of ICT can lead to continuous success.

FUTURE WORK

By the time simulation results are available the questions concerning evaluation and verification have to be answered. Being involved in several research projects dealing with emergency management in cooperation with many of the relevant organizations in that field it will be possible to verify the simulation model by performing field test. Parallel to the simulation efforts on operation execution a framework for an automated information exchange between the levels of operations planning and strategy is developed. All together this shall provide a broad alternative approach for the efforts in disaster relief.

REFERENCES


BIOGRAPHY

GERALD LICHTENEGGER studied mechanical engineering and economics with focus on industrial engineering at the Graz University of Technology, Austria. Since 2004 he has been research assistant at the Department of Engineering and Business Informatics. His fields of work at the department are production planning and control and business informatics. His main research interests are process analysis and process optimization especially covering the aspects of IT. Gerald Lichtenegger further serves as commanding officer at a county capital’s fire department in which he is responsible for the issues of IT.

SIEGFRIED VOESSNER holds a doctorate in Engineering Sciences. After being a visiting scholar at the Department for Engineering Economic Systems and Operations Research at Stanford University (USA) he joined the Top-Management Consulting Firm McKinsey & Company in Germany. In 2003 he accepted an offer for a tenured professorship at the at the Department of Engineering-Economics at Graz University of Technology in Austria. Since 2004 Prof. Voessner has been chairman of the then established Department of Engineering and Business Informatics. His research interests are Business and Operations Planning, Business Modeling and Business Simulation and the corresponding IT-Systems with a special focus on mission critical operations.
DT-MANETs for Rescue Information Service

Julien Franzolini*  Frédéric Guinand  Damien Olivier
Université du Havre  LITIS
25, rue Phillipe Lebon  F-76085 Le Havre
firstname.lastname@univ-lehavre.fr

KEYWORDS
Delay Tolerant, MANet, Risk Management

ABSTRACT

In case of natural or industrial disaster, alert and communication systems become an essential tool for rescue services and a way to inform and to alert population. The traditional communication and alert systems need static infrastructure. Mobiles devices are now ubiquitous and offer a free support for a spontaneous network. The goal of our project is to perform a delay tolerant network based on mobile devices. In this network mobile nodes are able to transmit data only with their neighborhood, it is an ad-hoc network. It gives no guarantee that a path between any couple of stations can be found. The network used a flooding method to propagate data which is called DFCN -standing for delayed Flooding with Cumulative Neighborhood-. This protocol uses two thresholds for avoiding the well-known “broadcast-storm”. This flooding model gives the ability to self-adapt to the environment when the distribution of devices is unknown and not uniform. In this document we present an experimentation of DFCN on mobile ad-hoc network apply to the risk management.

INTRODUCTION

With more than 70 SEVESO 1 (EUROPEAN Community 1996) companies, Haute-Normandie Region is particularly concerned with industrial risk. When a disaster occurs, communication becomes an essential tool for all rescue agents and a way to alert civilians. Generally, communication or alert systems need fixed infrastructure and professional material, unfortunately they are not always usable when a crisis happens. Today, mobile devices are omnipresent in our life and provide an inexpensive support for a network. We develop another communication support based on personal wireless mobile devices. DT-MANET stands for Delay-Tolerant or Disruption-Tolerant Mobile Ad-hoc Network, when mobile stations -like PDAs, Pocket PCs, SmartPhones...- are able to communicate only with their neighborhood.

The research partners are the Community of Agglomeration of Le Havre (CODAH) through the Board Information on Major Risks (DIRM) whose skills have turned to crisis management and the Computer Science, Information Processing, and Systems Laboratory (LITIS), which deals with computer aspects. Our view is to improve the reliability of warning systems and existing communications, using this technology as a complement to existing systems. The approach is not to replace the warning system sound (fire alarm) but allow better coverage of population and different actors, especially in cities where the density mobile devices is strong but where the propagation of sound waves is very disturbed. Moreover, the system could make on the ground greater responsiveness and richness of data between operational and destination of the population.

We have developed and implemented a particular routing method to propagate data. The protocol of communication is called DFCN (Guinand et al. 2006) - standing for Delayed Flooding with Cumulative Neighborhood-, it is based on the use of two thresholds for avoiding the well-known “broadcast-storm”. This protocol has the abilities to self-adapt to the environment when the distribution of devices is not uniform or unknown.

The paper is organized as follows. The following section gives a brief description of the Ad Hoc Network and focuses on the mobility in such networks. Then we present our contribution in the context of risk management. Finally discussion is developed and conclusions and planned future works are listed at the end of the paper.

AD-HOC NETWORK AND MOBILE AD-HOC NETWORK

A significant improvement in mobile stations is their ability to communicate through interface wireless technologies like Bluetooth and WiFi. In their current use, WiFi interfaces are used to connect to an access point offering a service (Internet, file sharing, etc. ...). This method of connection requires an infrastructure in opposition to the Ad-Hoc mode. Wireless interface are directly connected point to point in the Ad-Hoc mode without going through a fixed installation. In the con-
text of risk management, this mode of communication does not depend on any vulnerable infrastructure, which reduces sensitivity, increases strength and improves the scope of current systems alerts even the problem of autonomy subsists. According to standard 802.11 (ANSI/IEEE STD 802.11 1999) an Ad-Hoc network is a network of wireless devices connected to each other mode Ad-Hoc. This means that two stations can communicate if and only if they are both within their communication range, thus, a station can only communicate with devices located in its neighborhood (see figure 1). But recently, the concept of ad-hoc network has expanded considerably (Gerla et al. 2005) and we consider now in the following: Mobile Ad-hoc NETworks (MANET). These developments include:

Figure 1: Ad-hoc network according to standard 802.11

The belonging to a network. Contrary to the standard a mobile device must be in all station coverage to belong to the network. A connection between two nodes can be unidirectional. The network can be composed of subnetworks which aren’t linked together (see figure 2). The smallest subnetwork is only an unattached node so the minimum degree of connectivity is 0.

The mobility. In this network devices are susceptible to move in the environment, this mobility can have an impact or not on the network topology. This topology evolves in function of devices displacement and gives the dynamic of the network. (see figure 3)

The existence of a communication path. In a MANET, the connections between two nodes can be used to route data to other devices which aren’t in the same neighborhood. Some intermediate nodes are used as relay to propagate information. There are an important number of routing method in MANET but all depend on the topology of the network in other words of the environment: connectivity, devices density, maximum degree and the mobility model. This implies some research limitation in function of the desire goal.

Figure 2: Path and connectivity in MANet

DT-MANET stand for delay-tolerant or disruption-tolerant. It’s described a mobile ad-hoc network where the connectivity can’t be guaranteed. This situation occurs when the device density is not sufficient to guarantee the existence of a path between two stations (see figure 4). The routing method must be designed with the facts that the connexions can be lost at every time and that the presence or not of the destination node is not an error but a reasonable situation. (Jones et al. 2005)

With this hypothesis, to work correctly the routing method or protocol must be adaptive to the different network topology. It’s possible to have a part of the network with high density connect to an other part with only some connected devices.

Figure 3: Mobility in MANet

Figure 4: Current behavior in DT-MANet
These networks are characterized by:

- their robustness, DT-MANET don’t depend on infrastructure, and are based on simple rules. It gives fault tolerance. Lost a station doesn’t disturb the network, it is in fact its internal mechanic. In the case of an industrial or natural disaster, there is no dependency on radio relays, Internet and electricity. It gives robustness face on external events.

- their dubiousness, in this network a mobiles devices is not inevitably connected to another node and also a disconnection can occur at every time, this involves there is a doubt that data arrive to the destination.

NETWORK AND DATA MANAGING

The first step in our problem is to have a wifi support on mobile devices able to manage the network, on which we add a routing protocol or flooding protocol. All mobile implementations are designed for Windows mobile 2003, 5 and 6. These personal devices used the technology Net and are programed in C#; they need a specific library Open Net CF (OpenNETCF Consulting 2008) to obtain setting and tuning of the wifi interface. In ad-hoc mode the settings must respect two facts:

- only one network name (ibssid).
- only one emission frequency (channel).

Computers running on Linux are used to monitor the network. They are programed on Java and they don’t do anything in the network, they are not nodes.

Stations mobility and wifi layer

The wifi support layer must manage the network indeed the nodes movement implies a lot of connection and disconnection with their neighbors. In the wifi technology, this behavior is translated by synchronization and desynchronization. When this layer is active, it takes hand on the wifi interface. If the interface is already synchronized, the application leaves out the current connection and reconfigures the interface on the right frequency and network name. When a device encounters another station or a network, it begins a synchronization method.

The layer doesn’t only synchronize the interface although it can monitor the interface and retrieves data from wireless interfaces like: network reception sensibility, emission force, connection state.

A message validity management is included to the layer, the message header encapsulates some information like message id, sender id and validity date. If data are updated validity date changes and the message will be erased of the network. The last functionality is the neighborhood discovery, this process uses “Hello Hop” messages which contain the node id, “Hello Hop” are sent by all devices in regular intervals. For each “Hello Hop” interception, the id is added to the neighborhood list. If no discovery messages of a know station are received the node is deleted from the neighbors list. The neighborhood is 1-hop 5.

DFCN experimentation

The first protocol applies on our layer is an adaptive flooding protocol called DFCN (Guinand et al. 2006). This flooding model is based on traditional broadcast method, but has the particularity to take care of its neighborhood before sending data. This principle gives the advantage to avoid the broadcast storm which arrived when all devices want to send in the same time.

DFCN protocol needs several data about the device on which it is launched (m message id and s device id):

- DFCN must know its direct neighborhood called 1-hop : \( N(s) \).
- Each message includes a list of the neighbor which have already send the message : \( T(m) \).
- For each data, the device must have the message \( m \), the list of \( i_d \) which have received the message \( K(m) \), a boolean \( a(m) \) which indicates if the message must be sent or not.
- A random delay is added before sending to avoid the packet collision (Guinand et al. 2006) it is called RAD (Random Assessment Delay) \( r(m) \).

\[
\text{Algorithm 1: Algorithm executed upon message reception} \quad m : \text{The incoming message} \quad s : \text{The node which have sent} \\
\begin{align*}
\text{Data: } & m : \text{The incoming message} \\
\text{Data: } & s : \text{The node which have sent} \quad m \\
\begin{align*}
\text{begin} \\
\quad \text{if } m \text{ is received for the first time then} \\
\quad \quad K(m) \leftarrow T(m) ; \\
\quad \quad rad(m) \leftarrow \text{random} \in [0, \text{maxRAD}] ; \\
\quad \quad \text{else} \\
\quad \quad K(m) \leftarrow K(m) \cup T(m) \cup s ; \\
\end{align*}
\end{align*}
\]

\[
\text{Algorithm 2: The decision function defines if a given message is worthwhile to be forwarded or not : } a(m) \quad m : \text{The broadcast message} \quad m, \text{ candidate to immediate emission} \quad s : \text{The node that receives} \quad m \\
\begin{align*}
\text{Data: } & m : \text{The broadcast message} \quad m, \text{ candidate to} \\
\text{Data: } & s : \text{The node that receives} \quad m \\
\begin{align*}
\text{begin} \\
\quad \text{benefit} \leftarrow \frac{\vert N(s) \setminus K(m) \vert}{\vert N(s) \vert} ; \\
\quad a(m) \leftarrow \text{benefit} \geq \text{minBenefit} ; \\
\quad \text{if } a(m) \text{ then} \\
\quad \quad K(m) \leftarrow K(m) \cup N(s) ; \\
\end{align*}
\end{align*}
\]

518
Algorithm 3: The algorithm executed upon message reception.

Data: \( M(s) \) is the set of messages received and not expired by the node \( s \)

begin
  if \( |N(s)| < \text{densityThreshold} \) then
    foreach \( m \in M(s) \) do
      \( \text{rad}(m) \leftarrow 0 \);
  end

Experimentation show that DFCN protocol exhibits problem of saturate bandwidth when the number of nodes which have the message is important. Indeed this algorithm needs to propagate a list of already informed station and when the network is over 254 devices, the bandwidth is saturated by the list propagation. This list can have a larger size than the data themselves.

CURRENT WORK

For the previous point we work actually on an heuristic to reduce locally the list. The aim of this project is to design a complementary alert system and using the same network to route information and data to rescue teams. The first approach is to use wireless interface in ad-hoc mode to build a network which propagates data oriented to the risk management (Shibata et al. 2007).

The network support is now operational, it auto-configures the wireless interface without any extra human tuning on Wifi capable devices. After the network building, the data can be propagated with traditional broadcast or DFCN (The flooding method must be chosen before starting the application). This middleware layer has been tested on 10 mobile devices and monitored by subnotebooks running under Linux where we can see the local network topology. These subnotebooks are not in the network, they only grab wireless packets and extract information.

We finish a risk management application based on this network which is able to offer a sharing data service with localization. Each rescue agent has a mobile device wifi capable, he belongs to a group (Policemen, firemen ...). Each material contains a map of the site. Each participant moves within the environment and may take pictures, annotates the map and records audio comment. All Neighbors receive this information and may read if they belong to the right group and in all case relay the message according to DFCN rules.

CONCLUSION

DT-MAANets offer new inexpensive possibility for various application, but it is a challenge for computer science. The research on MANet is extremely different according to data are broadcasted, routed or locally used. Here we propose the use of an efficient broadcast protocol to alert and share information during a crisis. We project to have an approach less MANet with the use of fixed autonomous relays to diffuse information where the density of mobile is not sufficient. We can think that this system could be operational for large disaster, a tsunami for example to permit rescue services to share first urgency information.

REFERENCES

ANSI/IEEE STD 802.11, 1999. Wireless lan medium access control (mac) and physical layer (phy) specifications.


THE SIMULATION OF THE PROCEDURE FOR MULTIATTRIBUTE CHOICE OF THE STRUCTURE OF THE POWER PLANTS IN THE POWER SYSTEM

Eugeniusz M. Sroczan
Institute of Electric Power Engineering
Poznań University of Technology
ul. Piotrowo 3A
60-965 Poznań, Poland
E-mail: eugeniusz.sroczan@put.poznan.pl

KEYWORDS
Computer aided analysis, energy management, decision support system, interactive simulation, optimization.

ABSTRACT
The structure of sources committing in the power system (PS), depends on energy policy and affects the generation costs in the given power network. The expected costs of energy generated, in the examined PS for given time horizon, are calculated by using some simulation procedures. The results support the strategy operation, for the wholesale providers and power plants manage, assuming the varied structure of power plants and variable load of PS.

An example set of the power plants, including some types of primary energy sources: the hydro-power plants (HPP), pumped storage hydroelectric plant (PSHP) and wind power plants (WPP), is described accordingly in order to meet the balance of the generation and load in the PS. The developed simulator enables the optimization of the electric energy cost by using some procedures that assess load of sources committed in the discussed PS. The structure of the simulator (Sroczan 2005; Sroczan 2007) consists of some procedures which simulate the generation policy and check the economic quality of balancing the generation and demand of electric power and energy by using the fuzzy LP technique.

INTRODUCTION
The essential problem of local energy market (LEM) is to balance the energy production with consumers’ demands, with regard to the technical and economic boundaries, as well as legal restrictions outlined by EC law in the area of environment preservation.

Flow of power in PS, as certain of essential limitations in PS grid, is appointed at the same time as the time schedule of the power plants load (with graphic of loads) generated for power plants in accordance with momentaneous power demand $P_{Sg}$ of energy users. The proper power flow in PS grid enables the transmission of generated electric energy $P_{Sg}$ from source nodes to consumers of generated energy. If the flow is optimal any change of load causes additional losses of power and energy in the power line. In this manner the efficiency of conversion of the primary energy into electric one is decreased. The structure of generation set of committing types of power plants – affects the economy of generation due to possibility of covering constant or varied load. In case where the power of the given source depends on wind speed or water inflow, the time schedule programme of load must take into consideration some uncertainty of disposed volume of power or energy. The lack of power in that source node is compensated by additional flow from neighbour nodes, but this kind of possibilities depends on the structure of generation and power system reserve.

An operator of the power system is responsible for the realization of that program with regard to the frame work of sources of electric energy and transmission grids.

The interests of energy consumers are assured in the mode of the competition among the producers of energy as well as distribution companies at the wholesale energy market.

A local energy market LEM allows the producers and providers selling, both the electric energy and heat, to consumers allocated near to the plant of generation of the combined heat and power (CHP). The sources of type CHP due to integration of the processes of generation electric power and heat need, at the same time, both types of electricity and heat energy consumers (Jasiński and Kaproń 2007). They are causing the disturbances of electric power flow which is balanced in the local power system, expecting a varied constraints of heat demand (Kaproń and Kaproń 2007; Wydra 2007).

Regulations which are introduced by EU directives additionally involve the optimization of energy cost due to limiting constraints – the volume of energy generated from renewable sources of energy should approximate 7,5 % - 12% in the near future and up to 20% in the year 2020. This is an administrative boundary which should be taken into account in the policy of PS structure development (Sroczan and Urbaniak 2001).

The main aim of this paper is to simulate the effect of the structure of the electric power sources set in order to minimize the costs of energy with respect to given level of RER renewable energy ratio (Sroczan 2005) and development of the system as well as the modernization of the power stations. The additional constraints are connected with different customer behaviors. The proposed attempt is
based on the algorithm considering the real costs of energy in the given circumstances calculated with regard to demand and possibility of loading the renewable energy sources. The costs are calculated for hydro and wind power plants as well as for thermal fired with gas and coal. Finally results define the optimal policy for developing and reengineering the generation structure for the given PS containing different kinds of power plants.

**CALCULATING OF THE POWER LOAD FOR COMMITTING SOURCES OF ELECTRIC ENERGY**

The structure of PS is important in a case of failure or inconvenient weather conditions, for example too small or too great wind velocity for WPP or weak rain and flows in the hydropower plant. Electric power demanded by final energy consumers reflects the changes in the level of generation, transmission and distribution cost (Sroczan 2007). Assuming constant configuration of network, in the discussed PS, the costs of transmission and distribution are negligible. The cost of power and energy containing varied and fixed cost of generation is increasing due to the internalization of outside costs.

Energy costs include fuel burned – delivered cost of fuel (base cost, escalations, premiums or/and penalties, transportation, demurrage), outside laboratories fees and other outside costs related to fuel procurement. Additionally the cost of generated energy is increased by the payment for over limit (standard) emission of CO2.

In the work (Sroczan 2007) the PS operator, realizing the EU directives, defines the assumed value of the load covered by ecological sources using the preferred value off RER – renewable energy ratio defined as contribution of renewable sources to the PS generation process:

\[
\text{RER} = \frac{A_{RE}}{A_{Th} + A_{RE}}
\]

(1)

where: \(A_{Th}\) – volume of energy generated by thermal units fired with coal and gas, \(A_{RE}\) – volume of energy generated by power plants supplied with renewable energy.

In the presented attempt the RER value takes into account the disposed volume of renewable sources in discussed PS and at the same time enables the possibilities of changing the structure of generation. In this approach the volume of energy generated by all committing power plants, using renewable primary energy, is calculated as:

\[
A_{RE} = A_{PSST} \cdot \text{RER}
\]

(2)

where: \(A_{PSST}\) – volume of energy generated in the discussed PS during time period of optimization the cost of generation the electric energy.

**ATTRIBUTES OF POWER PLANTS COMMITTING IN THE PS**

The set of committing units should contain kinds of units (Sroczan 2005) which are characterized with following attributes, describing the operational parameters:

- units running with constant value of load \(P_g\) – like nuclear and thermal power plants;
- units covering the peak load of \(P_S\) – frequently realizing the start and shut down operation – like hydropower plants with limiting production;
- units enabling short time of charge - realizing some system services connecting with forced load gradient;
- power plants supplied with primary renewable energy (biomass, wind and hydro plant).

For the thermal power plants fired with coal the legally allowed levels of CO2 emission should be considered in the calculating processes.

The above mentioned attributes of power plants are considered in the process of optimization of the cost of generated electric energy.

**THE PROCEDURE OF SIMULATION AND OPTIMIZATION THE GENERATION COST OF ELECTRIC ENERGY**

The process of optimization of the work of committing power plants considers time period \(T\) in which all of the discussed units change the value of generated power with accordance to power demanded by end users. The goal is defined as minimization of generation cost with respect to the constraints.

\[
\mu_G(x) = \begin{cases} 
0 & x_i \leq c_i \\
1 - \left(1 + [d_i(x_i - c_i)]^2\right)^{-1} & x_i > c_i
\end{cases}
\]

(3)

\[
\mu_F(x) = \begin{cases} 
0 & x_i \geq b_i \\
1 - \left(1 + [x_i - (b_i - c_i)]^2\right)^{-1} & x_i < b_i
\end{cases}
\]

(4)

where: \(\mu_G(x)\), \(\mu_F(x)\) member functions for goal and constraints, \(d\) – parameter; \(c\) and \(b\) – values of constraints.

The value of \(x\) is normalized as \(L_x \|x\|_x = \max\{x_i\}\). The example results are shown in the figure 1.

Figures 1: Member function for goal and constraints function optimising the power dispatch.

The optimal committing of power plants is fulfil in the case:

\[
\min \{\mu_G(x), \mu_{c1}(x), \mu_{c2}(x) \ldots \mu_{cw}(x)\} \geq \lambda
\]

(5)

where:
\[
\lambda - 1 + \frac{b_i}{d_i} - \frac{(A^T x)_i}{d_i} = 0
\]  

(6)

\(\lambda\) – measure of the infringements the value of the goal or limitations; \(d\) – arbitrary measure of the non-execution of the given condition; \(A^T\) – matrix of constraints including the goal function; \(b^T\) – right side of the matrix of constraints including the expected value of goal function.

Over a longer period of time the simulation procedure allows the definition of the optimal policy of extending or modernization of the power structure with a regard to the constraints and regulations given by the energy and economy experts.

Optimal relationships among the stated power of all the sources depend on both the power source set and consumers’ behavior.

The developed simulator generates the range of acceptable values of RER coefficient balancing the load of committing units with regard to volume of generating electric energy using ecological sources of primary energy.

REFERENCES

Jasiński P. and H. Kapron. 2007. „Optimization of heat and power plant work in limit competition conditions”. 
Rynek Energii No. 2(75) ISSN 1425-5960 (in polish), 24-30.


EUGENIUSZ SROCZAN is employed as an assistant professor at the Poznan University of Technology (PUT) and professor of State Higher Vocational School in Gniezno. He 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 from PUT. 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 book on contemporary electrical installations of home. Since 2005 he has been the head of The Institute of Computing Science in State Higher Vocational School in Gniezno and since 1984 he has been the President of Branch of Polish Electricians Society at the PUT.

Figures 2: The result of Power dispatch among the committing power plants.

Analysis of cost of generated energy, realized with the use of developed simulator allows drawing of the following conclusion. The participation of power plants supplied with renewable energy sources cause the increase of the cost of generation in the discussed PS, due to additional costs stated by the rules of energy managing and these costs are transferred to the end users of energy.

Figures 3: The graph of loading the power plants as function of load increase.

This kind of the power plants run in the base layer of the load of power system load (fig 2. and fig. 3.).

CONCLUSIONS

The effect of PS sources structure on generation costs of electric energy is solved using dedicated procedures of simulation. The developed procedures enable calculation of the expected costs of energy for the given set of committing power plants and providers, assuming the varied structure of primary energy sources, in the meaning of types of power plant.
MODELING OF THE VULNERABILITY RELATED TO THE DYNAMIC ROAD TRAFFIC

Cyrille Bertelle  Antoine Dutot  Michel Nabaa*  Damien Olivier
Université du Havre
LITIS
25, rue Philippe Lebon
F-76085 Le Havre
{Cyrille.Bertelle; Antoine.Dutot; Michel.Nabaa; Damien.Olivier; }@univ-lehavre.fr
Pascal Mallet
CODAH
19, Rue Georges Braque
F-76085 Le Havre
Pascal.Mallet@agglo-havraise.fr

KEYWORDS
Dynamic Vulnerability Map, Decision Support System, Risk, Dynamic Graph, Communities Detection, Self-Organization, GIS, Traffic Flow

ABSTRACT

The utilization of the road network by vehicles with different behaviors can generate a danger under normal and especially under evacuation situations. In Le Havre agglomeration (CODAH), there are 33 establishments classified SEVESO with high threshold. The modeling and assessment of the danger is useful when it intersects with the exposed stakeholders. The most important factor is people. In the literature, vulnerability maps are constructed to help decision makers assess the risk. These maps are based on several types of vulnerability: socio-demographic, biophysical and other different types of hazards. Nevertheless, such approaches remain static and do not take into account the population displacement in the estimation of the vulnerability. We propose a decision support system which consists in a dynamic vulnerability map based on the difficulty to evacuate different sectors in Le Havre agglomeration. This map is visualized using the Geographic Information System (GIS) of the CODAH and evolves according to the dynamic state of the road traffic through a detection of communities in a large graph. This detection is realized by an ant algorithm.

INTRODUCTION

Le Havre agglomeration is exposed to several types of natural and industrial hazards: 16 establishments are classified SEVESO (EUROPEAN Community 1996) with very high threshold. The examination of impacted populations remains a difficult exercise. In this context, the Major Risk Management Direction team (DIRM) of Le Havre Agglomeration (CODAH) has developed a model which estimates the nocturnal and diurnal exposed population allocation PRET-RESSE (Bourcie and Mallet 2006); the scale is the building. Although the model is able to locate the diurnal and nocturnal population, it remains static because it does not take into account the daily movement of people and the road network utilization. For a better people evacuation in a major risk case, we need to have detailed information about the state of road traffic network to determine how to allocate the vehicles on the road network and model the movement of these vehicles. In fact, the panic effect of some people and the redundant drivers behaviors can lead to accidents and traffic jam, this can be very grievous and spread quickly.

In the literature, several models were developed to calculate a score for the vulnerability related to the road network utilization. Most of these models adopt a pessimistic approach to calculate this vulnerability: this case is met when a group of people in a hazardous area decide all to take the same route to evacuate this area, which unfortunately happens quite often in the real world evacuation situations. Although it helps decision-makers to estimate the risk by a census vulnerability map, this approach remains static and does not take into account the evolution of the road network traffic.

In this paper, we propose a dynamic and pessimistic approach related to the road network utilization. To this end, we model the road network by a dynamic graph (the dynamics is due to the traffic evolution). A simple model based on traffic flow will also be proposed and the interaction between micro and macro traffic simulation will be discussed. Then, we apply a self-organization algorithm (belonging to the collective intelligence algorithms) to detect communities in the graph. These communities correspond to clusters in the graph which evolve along the time. Each community is

*Corresponding author. Authors are alphabetically ordered. Michel Nabaa research is supported by Haute-Normandie Region.

1Directive SEVESO is an European directive, it lays down to the states to identify potential dangerous site. It intends to prevent major accidents involving dangerous substances and limit their consequences for man and the environment, with a view to ensuring high levels of protection throughout the Community.
a group of people in a neighborhood who share totally or partially, at a give time, the same route. The algorithm allows us to define the different vulnerable neighborhoods of the agglomeration in the case of a pessimistic evacuation due to a potential danger, while taking into account the evolution of the road network traffic due to the impact of the other flows coming from other neighborhoods. The result of this algorithm will be visualized into a GIS on a dynamic vulnerability map which categorizes various sectors depending on the difficulty of access to the road network. Finally, this vulnerability map will construct a tool to aid decision-makers in a better estimation of risk in the communes of the CODAH. This tool enriches the PRET-RESS static model developed at the CODAH, taking into account the mobility of the population.

STATE OF THE ART

Traditional methods evaluating the risk for population do not treat generally the behavioral of evacuee (e.g. initial response to an evacuation, travel speed, family interactions / group, and so on); they describe prescriptive rules as the travel distance. These traditional methods are not very sensitive to human behavior for different emergency scenarios. The computerized models offer the potential to evaluate the evacuation of a neighborhood in emergency situations and overcome these limitations (Castel 2006). Recently, some interesting applications have been developed by including the population dynamics, the models of urban growth patterns and land use.

For computer modelers, this integration provides the ability to have computing entities as agents that are linked to real geographical locations. For GIS users, it provides the ability to model the emergence of phenomena by various interactions of agents in time and space by using a GIS (Najlis and North 2004). So, combining several layers as houses, road network, population... allows us to model different types of agents into a GIS environment.

In (Cutter et al. 2000), the authors present a method to spatially estimate the vulnerability and treats the biophysical and social aspects (access to resources, people with evacuation special needs, people with reduced mobility ...).

Several layers are created in the GIS (a layer by a danger), and all these layers are combined into one composed of intersecting polygons to build a generic vulnerability map. To complete this, it was necessary to take into account the infrastructure and various possible routes of evacuation. So, a new map has been constructed and a new layer has been incorporated. This work has been applied to the George Town canton in which we find various natural and industrial risks, and where there are different types of people.

In the neighborhood evacuation cases on a micro scale, a number of studies based on micro simulation have been developed. In their paper (Church and Cova 2000), the authors presented a model to estimate the necessary time to evacuate a neighborhood according to the effective of the population, the number of vehicles, roads capacity and the number of vehicles per minute. The model is based on the optimization in order to find the critical area around a point at a potential danger in a pessimistic way. This model has been coupled with a GIS (ArcInfo) to visualize the results (identify evacuation plans) and construct an evacuation vulnerability map for the city (Santa Barbara).

Cova and Church (Church and Cova 1997) opened the way on the study based on geographic information systems to evacuate people. Their study identified the communities that may face transport difficulties during an evacuation. Research has modelled the population by lane occupation during an evacuation emergency using the city of Santa Barbara.

An optimization based model (graph partitioning problem) was realized to find the neighborhood that causes the highest vulnerability around each node in the graph and a vulnerability map around nodes in the city was constructed. A constructive heuristic has been used to calculate the best cluster around each node. This heuristic was developed in C and the result was displayed on a map (with ArcInfo).

Nevertheless, in this approach, we define the maximum number of nodes in a neighborhood, which may not always be realistic and does not take into account the traffic evolution during the calculation of critical neighborhoods. So, the vulnerable neighborhoods don’t evolve according to traffic state.

Since an individual panic under evacuation situation may cause a collective panic and the changes that may occur in the environment (buildings collapse, reverse route direction), the evolution of the system is unpredictable; so, in our work, we try to build a dynamic vulnerability map evolving with the traffic dynamics, in which the nodes number of in a critical neighborhood, is not predetermined and can change depending on actual traffic state.

DYNAMIC MODEL

Problem description

In this paper, the term "vulnerability" depends on the access to the road network. To address this vulnerability, we have to finely represent the population and the dynamic state of road traffic. In PRET-RESSE model developed within the major risks management team of CODAH, we have ventilated the day / night population at the building scale. The model was able to locate people during the day both in their workplace and their residence (the unemployed and retirees). It has been estimated that people will be in their residence during the night.

PRET-RESS will be enriched by our model that will try to dynamically assess the vulnerability related to the road traffic evolution; so we are interested in the allocation of the vehicles on the road network, the regulation of the traffic flow and trying to dynamically estimate the vulnerability according to the traffic.
System architecture

Our system consists of two modules as shown in the figure 1.

Figure 1: System architecture

The simulation module contains three components:

- The dynamic graph extracted from the road network layer and detailed in the following section,
- The flow management component consists of vehicles flow simulator applied on the graph,
- The communities detection component, detailed in the "Communities Detection" section. Its input is the extracted graph and the current flow. It returns the communities that are formed according to the current state of road traffic.

The visualization module consists of the road network layer integrated into the GIS. This module communicates with the simulation module: the graph is constructed from this module, which in turn get the simulation result and visualize it as a dynamic vulnerability map.

Environment modeling

The road network is integrated as a layer in the Geographic Information System (GIS). From this layer, we extract the data by using the open source java GIS toolkit Geotools. This toolkit provides several methods to manipulate geospatial data and implements Open Geospatial Consortium (OGC) specifications, so we can read and write to ESRI shapefile format. Once data road network are extracted, we use the GraphStream tool (Dutot et al. 2007) developed within LITIS laboratory of Le Havre to construct a graph corresponding to the GIS network layer. This tool is designed for modeling; processing and visualizing graphs.

Figure 2 represents the network layer into the GIS and the figure 3 is the graph extracted with geotools and visualized with GraphStream by using the same points coordinates.

The data extracted from network layer contains the roads circulation direction, roads id, roads type, their lengths and geometry.

The extracted multigraph $G(t) = (V(t), E(t))$ represents the road network at time $t$ where $V(t)$ is the set of nodes at $t$ and $E(t)$ the set of arcs at $t$. We deal with a multigraph because we have sometimes more than one oriented arc in the same direction between two adjacent nodes due to multiple routes between two points in Le Havre road network. GraphStream facilitates this task because it is adapted to model and visualize multigraphs. In the constructed multigraph:

- The nodes represent roads intersections,
- The arcs represent the roads taken by vehicles,
- The weight on each arc represents the needed time to cross this arc, depending on the current load of the traffic
- Dynamic aspect relates to the weights of the arcs, which can evolve in time, according to the evolution of the fluidity of circulation.

We have also constructed a Voronoi tessellation (Thiessen polygon) around nodes and projected the population in buildings on these nodes. The population in buildings is extracted from PRET-RESS model.

Vehicles flow

For a better people evacuation in a major risk situation, we need to know the state of the road network to determine how to allocate the vehicles on this network and to model the movement of these vehicles. Different types of models can be adopted:

- The microscopic model details the behavior of each individual vehicle by representing interactions with other vehicles and in general by using a spatialization. It is used on the scale of a sector or a neighborhood. It has the advantage to model vehicle behavior in an evacuation of a neighborhood, people panic, interactions between vehicles, accidents... So, if we have vehicles with different behaviors and interacting between them, these vehicles sometimes self organize and with a top down approach, we can examine the global behavior of the system and try to locally modify the environment, when necessary, to ameliorate the system,
- The macroscopic model is based on the analogy between vehicular traffic and the fluid flow within a canal. It allows us to visualize the flow on the roads rather than individual vehicles. It is used at many sectors or the entire city scale,
- The hybrid model allows coupling the two types of dynamics flow models within the same simulation. Several works have already borrowed this direction (Hennecke et al. 2000, Bourrel and Henn 2002, Magne et al. 2000), however, this approach is relatively new and very few have adopted it to our knowledge (Hman et al. 2006). The use of a hybrid model is very important to us: changing the scale from micro to macro in a region where we haven't a crisis situation (everything is normal) allows to economize the computation and the
change from macro to micro in a critical situation allows to zoom and detect the behaviors and interactions between entities in danger.

In this paper, we used a simple model of macroscopic flow:

- A flow of cars moving from one arc to another adjacent one,
- The arcs are limited in capacity,
- The flow can be broken and two or more flows can gather on a node,
- Traffic jams may appear in certain places of the road network; those places will be more vulnerable than others.

We have adopted a macroscopic model in which flows circulate normally (no accidents) because the goal now is to establish a dynamic pessimistic vulnerability map which is not always the case in the real world (e.g. 90% of people take an exit route and the rest takes another route for example). Hence, it is important to have in the near future a micro approach with a change of scale (from micro to macro and vice versa during the simulation) to simulate scenarios of danger in real time (accidents, behavior of drivers, vehicles interactions ...), a study on which we are working actually.

**COMMUNITIES DETECTION**

Our aim is to identify communities in graphs, i.e. dense areas strongly linked to each other and more weakly linked to the outside world. If the concept of communities in a graph is difficult to define formally, it can be seen as a set of nodes whose internal connections density is higher than the outside density without defining formal threshold (Pons 2005). Thus the goal is to find a partition of nodes in communities according to a certain predefined criteria without fixing the number of such communities or the number of nodes in a community. Interesting works were developed in the literature on the detection of structure in large communities in graphs (Clauset et al. 2004, Newman 2004a;b, Pons and Latapy 2006).

In our problem, we look for a self-organization in networks with an algorithm close to the detection of communities in large graphs and belonging to collective intelligence algorithms. Organizations connect elements, events or individuals by interrelations so that they become components of a whole. They assume the solidarity and robustness of these links, and ensure that the system will eventually be long lasting despite random perturbations. The organizations, therefore: transform, produce, tie and maintain. Time is present generating the dynamic and we will try to fight against these organizations, in the case of risk management, to avoid bottlenecks which do not facilitate the evacuations.

Thus we will considered the graph at time $t$, $G(t) = (V(t), E(t))$ where the edges are weighted, this weight being noted $|e|$ for the edge $e$ and represents the needed time
to cross this arc, depending on the current load of the traffic and we try to define a colored dynamic graph\( G(t) = (E(t), V(t), C(t)) \).

The algorithm used can be referred to as an ant algorithm (Dorigo and Stützle 2004). Our ant algorithm use several colonies of ants, each of a distinct color. Ants travel inside the graph and lay down pheromones, information that can be detected by other ants. Pheromones are also colored. Ants tend to be repulsed by pheromones of other colors. Furthermore, ants tend to favor edges with important weights.

The colored dynamic graph precedent mentionned is defined such that:

- \( V(t) \) is the set of vertices at time \( t \). Each vertex \( v \) is characterized by:
  - a color \( c \in C(t) \),
- \( E(t) \) is the set of edges at time \( t \). Each edge \( e \) is characterized by:
  - a weight \( |e| \in \mathbb{N}^+ \) that corresponds to interaction importance between the elements at each end of edge \( e \).
  - a quantity of pheromones of each color.
- \( C(t) \) is a set of colors representing the ant colonies at time \( t \).

The algorithm principle is to color the graph using pheromones. Each colony will collaborate to colonize zones, whereas colonies compete to maintain their own colored zone (see figure 4). Solutions will therefore emerge and be maintained by the ant behavior (see figure 5). The solutions will be the color of each vertex in the graph. Indeed, colored pheromones are deposited by ants on edges. The color of a vertex is obtained from the color having the largest proportion of pheromones on all incident edges (see algorithm 1).

We have an interaction between each two local adjacent nodes according to the attraction force that exists between them. This force depends in our case on the report of the number of vehicles on the arc between 2 nodes neighbors / vehicles capacity of the arc. This report was chosen because, in every community, we will have a large number of vehicles.
which all decide to exit through a single road in the case of a potential danger; this responds well to one of the purposes listed in beginning to have a pessimistic approach in the calculation of vulnerability. The algorithm has the advantage of not allowing the breaking of a link between 2 adjacent nodes to maintain the structure of the road network. When the traffic evolves, the algorithm detects that and communities can change or disappear as a result of local forces that change between the nodes locally.

**Dynamic vulnerability map**

At each simulation time step, the flow on the arcs changes following traffic conditions and the attraction may change also. Once communities are formed on the graph, the result will be transmitted to the road network layer into the GIS to
CONCLUSION

In this paper, we have proposed a decision support system to assess the danger depending on the road network usage by the vehicles population using the network. This tool enables decision makers to visualize, on a geographic information system, a dynamic vulnerability map related to the difficulty of evacuating the various streets in the metropolitan area of Le Havre agglomeration. We simulated road network traffic by using a simple model of vehicles flow. A communities detection algorithm in the large graphs was adopted. It enabled us to form communities in a graph thanks to local forces propagation rule between adjacent nodes. The communities evolve according to the current state of the road network traffic. The result of the evolution of communities is visualized by using a GIS. The adopted approach allowed us to estimate the risk due to the use of the road network by vehicles and categorize Le Havre agglomeration areas by their vulnerability. We will complete our work by using real traffic data retrieved from a displacements survey with Le Havre population which will help us to better locate people during the day and therefore having a more realistic vulnerability dynamic map.

REFERENCES


SPECIFIC MODELS
SYSTEM DYNAMICS VERSUS CELLULAR AUTOMATA IN MODELLING PANIC SITUATIONS

Provitolo Damienne
Chargée de recherche CNRS
UMR 6049 ThéMA, CNRS, Université de Franche-Comté
32 rue Mégevand, 25 030 Besançon
UMR 6526 Géosciences Azur, CNRS, UNSA
250 rue A. Einstein, 06560 Valbonne
damienne.provitolo@geoazur.unice.fr

Daudé Eric
Chargé de recherche CNRS
UMR 6266 IDEES, Université de Rouen
7 rue Thomas Becket, 76 821 Mont Saint-Aignan
eric.daudé@univ-rouen.fr

KEYWORDS
panic, system dynamics modelling, cellular automata, simulation, emergence

ABSTRACT
In this paper we face system dynamics modelling and cellular automata to model panic processes. We compare both models using phase plans. The results of simulation confirm our hypotheses: First, a collective behaviour is not the arithmetical sum of the individual behaviours. Secondly the crowd causes the emergence of collective panic from individual panic. Both types of methodology produce the emergence of panic and identical curves for different values of initial conditions and parameters. But the effects of thresholds vary according to these values.

INTRODUCTION
Frequently, technological or natural disasters create panic behaviours. If these behaviours are not the most frequent, they are the most dreaded (Crocq 1994) and this for three reasons: they increase the number of victims, they are difficult to stop, and they have impact on disaster relief, and on the evacuation of the population.

The studies of the behaviour of panic during disasters are based essentially on the analysis of documents (archives, account of events, press cuttings) and on the observations realized by rescuers, medical profession and psychologists. Some agent based models study the evacuation of the panicking population (Helbing et al. 2000, 2002). These models are mostly applied to enclosed spaces. But few researches are interested in the processes of self-organization and in the emergence of the collective panic from individual panic.

In this paper, we compare two methods of simulation of the emergence of the collective panic from individual panic. These two methods are the system dynamics modelling and the cellular automata. The results of simulation based on these two approaches are different, although the structure and the hypothesis of the models are identical. We wish to defend the idea that the different results are not inevitably a limit of the modelling. According to the methodology and thus according to the scale of analysis of the phenomenon, the different results of simulation contribute to the knowledge of the emergence of panic.

UNDERSTANDING THE PANIC BASED ON INDIVIDUAL INTERACTIONS
The collective panic propagates in a crowd from behaviours of imitation (Dupuy 1991; Quarantelli 1954). It is indeed easier to follow his neighbours and the group than to think. In situation of stress, individuals who do not know which behaviour to adopt, adapt their behaviour according to their neighbours. In other words, the individuals imitate the others. The crowd is the support of contagion of the panic (Le Bon 2003). The collective panic emerges without concerted action. The collective panic would thus appear from the diffusion of individual panic, without a leader who would call to the panic, or prevent it.

We put forward three hypotheses to explain the propagation of the panic in a crowd:
Hypothèse 1: The panic spreads by behaviour of imitation. The imitation is based on the pattern of interactions between the individuals.
Hypothèse 2: A collective behaviour is not the arithmetical sum of the individual behaviours.
Hypothèse 3: The crowd causes the emergence of collective panic.

The models, presented in section Two models to simulate emergence and diffusion of panic are based on the epidemiological models of W. Kermack and A. McKendrick (1927). Three state variables describe the system of panic: the population susceptible to panic (Psp), the panicking population (Pp) and the non panicking population (Npp). These three state variables constitute the crowd, i.e. the total population. The total population is a fixed population of size N. The three populations (Psp, Pp, Npp) meet each other and interact. Transmission rate apprehend the contagion of the panic between both human populations in contact. Interactions between human populations indeed do not necessarily lead to contagion. This transmission rate is a coefficient which varies from 0 to 1, i.e. a low to a high contamination. After a certain period of time, people will stop panicking and resume normal behaviour. The non-panicking population is proportional to the numbers of panicking population and the return time to normal behaviour (Rtn).
TWO MODELS TO SIMULATE EMERGENCE AND DIFFUSION OF PANIC

We study the emergence and the spread of panic from two different methods of modelling: the System Dynamic (SD) and the Cellular Automata (CA). With the system dynamics modelling, the system in whole is the object of study. The systemic approach allows us to search the principles of organization and functioning of the system of panic. With cellular automata, individual behaviours and their interactions are the studied objects. The conditions of a possible organization and the relations between a micro level and a macro level are studied.

A system dynamics model of panic

The dynamic modelling of the system is found originally in the General System Theory of Ludwig Von Bertalanfy (Von Bertalanfy 1973). A system consists of several components connected by flows of material, energy or information. System dynamics is a methodology used to understand how systems change over time. The mathematical formalism of the system dynamics modelling is based on differential equations. The “Stella Research” software solves differential equations as difference equations. The graphical formalism is based on stocks, flows, converters and connectors. The stock or state variables are the reservoirs of the system. The flow variables correspond to the processes. Converters are rates; they connect different types of flows. Finally connectors are the relations and the feedback between the elements of the system.

Figure 1 shows the graphical formalism of panic model (Provitolo 2007) built with the “Stella Research” software. The model was presented during the European Conference Complex Systems -EPNACS’07. The aim of this paper is different. We compare the results of simulation obtained by both methods of modelling. The aggregate model includes three stocks of population: the population susceptible to panic (Psp) (1), the panicking population (Pp) (2) and the non-panicking population (Npp) (3), i.e. people will stop panicking and resume normal behaviour; interactions (4) between these three types of population, transmission rate of panic (Tr) and return time to a normal behaviour (Rtn). “Adoptions” (5) and “normal behaviour” (6) flows connect the three stocks of population. These three stocks constitute the total population (7).

The corresponding equations are:

\[
Psp (t) = Psp (t - dt) - \text{(adoptions)} \times dt \quad (1)
\]

\[
Pp (t) = Pp (t - dt) + \text{(adoptions} - \text{Normal behaviour}) \times dt \quad (2)
\]

\[
Npp (t) = Npp (t - dt) + \text{(Normal Behaviour)} \times dt \quad (3)
\]

\[
\text{Interaction PspPpNpp} = \text{FractionPsp} \times \text{Fraction Pp} \times \text{Fraction Npp} \times \text{Total Population} \quad (4)
\]

\[
\text{Adoptions} = \text{Interaction PspPpNpp} \times \text{Tr} \quad (5)
\]

\[
\text{Normal behaviour} = \frac{Pp}{Rtn} \quad (6)
\]

\[
\text{Total Population} = \text{Psp} + \text{Pp} + \text{Npp} \quad (7)
\]

Figure 1 A system of panic based on the interactions between populations

With the system dynamics (SD), we compute how population of agents behaves as a whole. With the cellular automata (CA), individual agents are not processed as a stock but as individuals.

A cellular-automata based model of panic

A second step of this research uses cellular automata to build the same model, i.e. a model of panic based on interaction between individuals.

We identify three main differences between system dynamic and cellular-automata. First is the spatial component of cellular-automata, defined in this model by a two-dimension grid of \( t \times n \) cells. This spatial structure defines local interactions in cellular-automata and not global as in system dynamic, which means that a cell is only connected with its neighbour’s cells, 4 (von Neumann neighbourhood) or 8 (Moore neighbourhood) for example. Second is the nature of the dynamic, performs by transition rules \( T \). In cellular automata, the future state \( S(t+1) \) (8) of one cell depends of its actual state \( S(t) \) and of its environmental state \( Ev \). We can then define cellular automata as a discrete and local dynamic system where:

\[
S(t+1) = T(S(t), Ev(t)) \quad (8)
\]

is the transition rule applied at each time and for all cells. A third difference is more implicit, it is based on the local and distributed nature of CA compared to the global and centralised nature of SD: we use to view cells as micro-structures or micro entities. In this way, it’s accurate to express this alteration by changing the name of the different states in the model. A cell represents a group of individuals which can takes three different states. The state space (9) is:

\[
S = \{Gsp, pG, npG\} \quad (9)
\]

where transition is not symmetric, \( Gsp \rightarrow pG \rightarrow npG \).
In this way, population of susceptible (Psp) is the sum of groups of susceptible (Gsp). Then and to be in accordance with the system dynamics model, we use three variables (10, 11, 12) to observe the dynamic of the model:

\[ Psp_i = \sum_{i,j} Gsp_{i,j} (t) \]  
\[ Pp_i = \sum_{i,j} pG_{i,j} (t) \]  
\[ Npp_i = \sum_{i,j} npG_{i,j} (t) \]

The environment of a cell \( c_i \) is composed of its 8 neighbours cells, where \( v \) (13) gives the coordinates of the cells centred on the cell \( c_i \):

\[ v = ((0, -1), (1, 0), (0, 1), (-1, 0), (-1, -1), (1, -1), (1, 1), (-1, 1)) \]

(13)

The transition from the state Gsp - the group susceptible to panic -, to the state pG - the panicking group -, depends on the proportion \( v_i \) of panicking groups in the spatio-temporal neighbourhood of the cell, which is the average of panicking groups in this vicinity (14):

\[ < v_i (t) >_{\chi \times \tau} = \frac{1}{n_R} \sum_{j \in V(i)} v_j (t) \]

where \( n_R \) is the density of all groups in the neighbourhood. As the global density is equal to \( h = n \) in this model, \( n_R \) is equal to 8 for each cell. This parameter \( v_i \) represents the probability to interact with a group of panicking people. But it is not sufficient to interact with a panicking group to be panicked. The transition probability from the state Gsp to the state pG for the cell \( i \) (15) is then:

\[ p(s_i = pG / s_i = Gsp) = t_r \cdot < v_i (t) >_{\chi \times \tau} \]

(15)

where \( t_r \) is a continuous parameter representing the transmission rate of the model, the probability that an interaction transmitted the panic. This transmission rate varies from 0 to 1. A main difference with the system dynamic model is the result of this process: the result is discrete (panicking or not) whereas with system dynamic the result is continuous (a proportion of panicking people at each round). With cellular automata, we then have to choose a way to transit from the state Gsp to the state pG, depending of this probability. We use the algorithm (16):

```
to Panic
if (pG / (Gsp+pG+npG)) * tr > random-floating l
[sist(Gsp) \rightarrow (pG)]
end
```

(16)

where random-floating reports a uniform random floating point number included in [0;1]. Once in the state pG, the group \( i \) stay in this panicking situation a certain laps of time \( Rtn \), and then pass on the state npG, the panic is off in this group and the transmission to the neighbours is off as well (17).

\[ s_i (t) = \begin{cases} 
  pG & \text{if } \tau_i (t) \in [0, Rtn] \\
  npG & \text{else}
\end{cases} \]

(17)

where \( \tau_i(t) \) represents the time passed since the panic is on, which is linked with the \( Rtn \) in the system dynamic model. We have tried a probabilistic method to smooth the transition between the pG to the npG but results do not give new interesting facts.

**SIMULATIONS TO PREDICT EMERGENCE AND DIFFUSION OF PANIC**

According to the values of initial conditions and of parameters, we observe qualitative modifications of the trajectories for both methodologies. On the other hand, according to the methodology, i.e. the system dynamics modelling or the cellular automata, the results differ for identical values. These results will be analyzed in the section Discussion.

**Results of simulation with system dynamics modelling**

We present results of simulation for different values of panicking population and of parameters (transmission rate and return time to normal behaviour) related to the model. We will focus more particularly our attention on the spread and the emergence of panic. The population susceptible to panic is always equal to 3721 individuals. The phase plans (Figures 2-3) allow us to analyse the various trajectories of the system.

Two cases are presented.

**Case 1**: the population susceptible to panic is equal to 3721 individuals and the panicking population is equal to 50 individuals. We study the system evolution by making vary the transmission rate of the panic (between 0 and 1, i.e. a low to a high contamination.) and the return time to normal behaviour (between 1 and 10). Nine tests are realized with values of \( Tr \) equal to 0.1; 0.5 or 0.85 and values of \( Rtn \) equal to 1, 5 or 10 (Figures 2a, 2b, 2c).

**Case 2**: the population susceptible to panic is equal to 3721 individuals, the panicking population is equal to 200 individuals. As in the case 1, we study the system evolution by making vary the transmission rate of the panic (between 0 and 1). The return time to normal behaviour is equal to 22 units of time (Figure 3).

**Case 1**:

- \( Psp = 3721 \)
- \( Pp = 50 \)
- \( Npp = 0 \) (at the beginning of the simulation, this stock is equal to 0 because the panicking persons have not yet found their normal behaviour)

Transmission rate \( (Tr) = 0.1; Tr = 0.5; Tr = 0.85 \)

Return time to a normal behaviour \( (Rtn) = 1 \) units of time, \( Rtn = 5, Rtn = 10 \)
population. On the other hand, for these same conditions, the results of simulation with cellular automata differ.

Case 2:
$Psp = 3721$
$Pp = 200$
$Npp = 0$

Transmission rate ($Tr$) = 0.1; $Tr = 0.5; Tr = 0.85$

Return time to a normal behaviour ($Rtn$) = 22 units of time

The trajectories converge to an equilibrium point (coordinated 0, 0) where $Psp$ and $Pp$ are zero (Fig. 3). This equilibrium point with coordinates (0, 0) can be explained by the flow normal behaviour which tends to empty the stock panicking population and to feed that entitled non-panicking population.

But contrary to the case 1, the collective panic emerge. The population susceptible to panic and the panicking population decrease at first, and then increase. There is a bifurcation and the emergence of the panic before reaching a new bifurcation bringing the system to another state of equilibrium where $Psp$ and $Pp$ are zero. The modification of the initial condition of the panicking population and $Rtn$ influence the proportion of population susceptible to panic and panicking population. There is a threshold, with $Pp = 200$, $Rtn = 22$ and $Tr \geq 0.5$.

**Results of simulation with cellular automata**

We use the same initial conditions as previously (case 1) with the system dynamic model, defined for each simulation by the set $\{pG; Rtn; Tr\}$. The domain is composed of 3721 cells in a regular grid.

Figure 4 shows two very different simulations with for the upper one the set of initial conditions equal to $\{50; 10; 0.85\}$ and the lower one initial conditions equal to $\{50; 7; 0.2\}$. These simulations show the importance of parameters in the spread of the panic, especially transmission rate.

This suspicion fits with the study of parameters space built with the set $\{50; 1,\ldots, 10; 0.05,\ldots, 0.85\}$ (Figure 5). When
transmission rate is lower than 15%, a high period of transmission cannot expand the volume of panicking population. On the contrary, when transmission rate is high, up to 70%, even a small period of panic activity at the group’s level produces a high level of collective panic. The most interesting zone here is the small part of the system where panic can go up or go down, in the area of 50% people panicked. A small difference in one or other parameter can make the system go from one attractor to the other.

Figure 5 Parameter space of panic’s diffusion: from low diffusion (yellow) to global diffusion (red)

<table>
<thead>
<tr>
<th>Init</th>
<th>3</th>
<th>7</th>
<th>11</th>
<th>12</th>
<th>16</th>
<th>21</th>
<th>28</th>
<th>35</th>
<th>42</th>
<th>55</th>
</tr>
</thead>
</table>

Figure 4 Phases space and screen-shot of 2 simulations

**DISCUSSION**

The study of panic enables to put human vulnerability in the foreground of the analysis of disasters. But the three hypotheses simplify the real situations. For example, the sensitivity of the population is not included in the model. This sensitivity depends on the age of the population, on the social structure (Granovetter 1978), or others factors. The models, in the same way, do not integrate the shape of the environment, for example the shape of building or of road.

Another limitation of this work is the quantitative prediction. We cannot predict the number of panicking population for three reasons: At first, experimentations in situ are few. These experimentations, when they are carried out, give an uncertain estimation of the behaviour that could be effectively chosen in the emergency of the situation. It is difficult to predict what will happen. Secondly, the data to calibrate the models do not exist. Thirdly, it is difficult to quantify, at the beginning of the event, the number of panicking population, population who contaminate the others.

These two methods give us a global understanding of the spread of the panic in the population. It is indeed possible to predict qualitatively the dynamics of the phenomenon and to identify the bifurcation. Panic is not a fate. Modifying the panic, and thus the vulnerability of the population, requires planning and preparedness: Implementation of a variety of measures such as prevention, evacuation before an event occurs, knowledge of the preventive measures in order to avoid the collective behaviour of panic.

The various methods of simulation show that different results are not inevitably a limit of the modelling. Several models represent the same reality (Israel 1998). And it is difficult to decide which of these two models give the best representation of the collective panic. According to the methodology and thus according to the scale of analysis of the phenomenon, the different results of simulation represent certainly a contribution of knowledge on the studied system. Whatever the methodology will be, the results confirm our hypotheses: First, a collective behaviour
is not the arithmetical sum of the individual behaviours. Secondly the crowd causes the emergence of collective panic from individual panic. There is indeed a repetition of the behaviour in the crowd.

Both types of methodology produce the emergence of the panic and the forms of identical curves for different values of initial conditions and parameters. But the effects of thresholds vary according to the values. For example, with cellular automata, the panic spread with 50 panicking population in a total population of 3721 people. With the system dynamics modelling, 200 persons are needed to observe the emergence of the panic.

A difference between the SD and the CA model is that $\tau(t)$ is turned down of one unit at each round, and a group quit the panic situation only if $\tau(t)$ is equal to zero. Whereas panicking is submitted to a probability, stop panicking is deterministic in this model. With the system dynamics modelling, at each step, the value of inflow is added to stock. For example, the value of inflow normal behaviour is always the previous value of stock panicking population times a coefficient, here the return time to normal behaviour.

CONCLUSION

The emergence of the panic does not appear in every situation. This emergence depends on three elements: the transmission rate, the return time to normal behaviour, but also the number of panicking population at the beginning of the simulation. The results of simulation of these two methods confirm our hypotheses: a collective behaviour is not the arithmetical sum of the individual behaviours and the crowd causes the emergence of collective panic from individual panic. To conclude, the models of simulation are a representation of the reality (Isråël 1998) and not the representation of the reality. Whatever the methodology will be, the modelling and the simulation of many agents in interaction ask the question of the emergent properties.

REFERENCES


An Architecture for Individual Behaviour Modeling in Emergency Situations

Julien Valentin(*)(**) Michael Bergeret(*) Eric Gouardères(*) Wilfrid Lefer(*)

(*LIUPPA, University of Pau (**CSTB, Sophia Antipolis, France julien.valentin@etu.d.univ-pau.fr

ABSTRACT

Nowadays, several computational tools for simulation and design of emergency egress (evacuation) are available. In this paper we evaluate the technical features of existing models, especially their reactive/cognitive aspects, and we propose a hybrid cognitive/reactive architecture for the simulation of human decision in emergency simulation.

KEYWORDS: Emergency egress, Human behaviour simulation, Decision model, Hybrid architecture, Multi-Agent Systems.

Introduction

Two aspects can be considered to adress the issue of the human behaviour simulation in egress situation. The first one is the psycho-social aspect, which refers to the influence of other persons (social aspect) and mental state (psychological aspect) on the individual behaviour.

The second one is the decision making aspect: simulating human behaviour requires to model perception, reasoning and decision making mecanisms.

This paper adresses the decision making process for displace-ment of autonomous avatars. Firstly, a technical overview of motion decision models of main existing simulation tools is presented. Then, we propose a new hybrid cognitive/reactive architecture to handle the psycho-social aspects of human dec-ision in egress situation.

Technical overview of existing egress simulation tools

Exodus [1] is cited as a typical example of an egress simu-lator based on a purely cognitive design. In this system, any movement is governed by assessments of move cost from a cell to the next. Heuristics and rules are used in order to achieve path resolution. There is no reactive motion decision model. This model is the most accurate decision-making system because pedestrians run again their reasonning at each time step.

Evac opposes fairly easily with the purely cognitive approach of Exodus. As the movement is governed by local rules, it represents a totally reactive approach of human behaviour. The work of Helbing [3] is integrated in order to assess physical interactions between avatars. In this system, the direction of movement is determined according to the output of a fire simulation tool, which provides a realistic simulation of the spread of fire and smoke. Crowd is seen as an incompressible fluid and doors as extraction devices. The spread of this fluid is simulated using a fire dynamics tool, which generates a field of vectors that can be used for influencing decision making of individuals during the simulation. The lack of a cognitive model is the main drawback of this system.

HiDac [4] combines a low level motion decision model, a high-level decision model and an emotion decision model (PMFServ), strongly coupled together. The low-level (reactive) decision model is based on a mix of Reynold’s local rules system and a weighting system based on others continuous rules. Behaviours can be blended so as to generate new behaviours. Discrete rules are used to avoid shaking problems due to discrete time integration. The high level (cognitive) decision making system is based on a static tree decision making and implements the path planning, which is not described in details but appears to be a dynamic path planning model. The main lack of this system is that there is no hierarchical decomposition of the motion decision model. Massegress [2] was created by Xiaoshan Pan [5] in order to validate his works on human behaviour terminology during emergency egress. He introduced the notion of behaviour adaptability during egress situations controlled by stress and also a panel of atomic behaviours based on reynolds rules: herding, queuing, competitive, etc.

The behaviour adaptability is made from a decision tree, which elects a behaviour by considering local perception of the individual but also his/her internal state. Decision making is described upon this static decision tree but the path planning is defined as local rules, which yields to a totally reactive process, which lacks of a reasoning approach.

<table>
<thead>
<tr>
<th></th>
<th>Reactive Model</th>
<th>Cogntive Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exodus</td>
<td>None</td>
<td>Dynamic planning Rules base</td>
</tr>
<tr>
<td>EVAC</td>
<td>Equation based</td>
<td>None</td>
</tr>
<tr>
<td>Hidac</td>
<td>Local rules parametrized</td>
<td>Decision tree Dynamic path planning</td>
</tr>
<tr>
<td>Massegress</td>
<td>Local rules</td>
<td>Decision tree Local navigation</td>
</tr>
</tbody>
</table>

Figure 1: Reactive and cognitive aspects of current emergency egress systems.

The analysis of these plateforms highlights the need of an evoluted decision making process. A static decision tree gives a meta-model for the decision process of the
individuals but does not provide the flexibility of rule based systems. A reactive model is very useful to simply solve conflicts of position before dealing with physical collisions.

The next purpose in this paper is to put decision model, psychological, and social adaptability, all together, thus associating a reactive model, a planning model and goal directed decision mecanism in a hierarchical architecture.

A new architecture for emergency behaviour modeling

Hybrid architecture

Taking into account both the environment and individuals internal data requires a two-level decision making process. The reaction level takes decisions only based on environment data, which corresponds to a reflex type behaviour. The second level is able to adapt the decision to the mental states of the individual and thus could be characterized as cognitive behaviour. In multi-agent systems, there is a type of architecture which have both a cognitive and a reactive decision-making part, called hybrid architecture. Most of those hybrid architectures are layered. They can be classified in two categories [6]: those made with horizontal layers, such as TouringMachines [7], and those based on vertical layers, such as InteRRaP [8]. In horizontal layers ones, data from the environment are processed by each layer in order to have multiple possible answers and then a control structure selects the most appropriate of these answers. The chosen response is the next action to process. In vertical layers architectures, the decision making process is serialized, which means that a unique answer is computed. Vertical layers architectures are generally preferred for real time simulations because they are more efficient and hence we have adopted this approach. A crucial concern of these hybrid, layered architectures is the interface between the cognitive and reactive layers, for which Andriamasinoro [9] identified some potential solutions.

Our proposal

Figure 2 shows our architecture for decision making. It is based on InteRRaP [8] and works as follows. The behavioural layer tries to find out a way to respond to the actual event; if it fails, the planification layer attempts to provide a solution and, in case of failure, the cooperative layer has to solve the problem. This functional scheme is based on the following considerations. The more knowledge the decision process takes into account, the better the solution is, but the less layers are involved in the decision process, the more efficient it is. So the challenge here is to find the better tradeoff between solution accuracy and computational time.

Layers descriptions

Behavioural layer

The behavioural layer contains two types of decision patterns:

- a reactive one which has only access to the percepts
- a goal oriented one which may exploit the model of the world, mental map and internal state.

They are gathered in a subsomption architecture to elect the more appropriate. Typical behaviours are: follow a path, avoid collision, etc. A particular behaviour is panic, which is triggered when the stress level is above a certain threshold.

Planning layer

The first cognitive layer is the planning layer. Here behavioirs are elaborated from agent knowledge, including mental representation of the environment, internal state, and percepts data. They are built from identified plan patterns, such as path finding, help somebody, etc, which can be combined together to elaborate more complex behaviours. As for the behavioural layer, a particular module handles emergency situations, which are triggered when the stress level is above a certain threshold.

Cooperation layer

This layer is in charge of agents’ interactions and cooperation. In addition to lower level layers knowledge, agents’ acquaintances are taken into account. Cooperation patterns and interaction protocols are defined.

Functionnal aspects and stress management

In our model, the access to the upper layer can be inhibited using a stress threshold permitting user to describe emergency behaviors (eg: Panicking at behavioral layer and Explore at the Planification one). This specificity of the architecture permits to introduce two decision models for the plan-
ification and behavioural layers: when the stress condition failed to propagate percepts to the higher layer, an emergency decision module produce a rather urgent decision according the layer it’s concerned by. Figure 3 shows an concrete illustration of how the architecture works through samples behaviours described above. It can be noted that in this sample no goal selection is necessary because the only goal is to find a way out of a building in emergency situation.

![Architecture Diagram](image)

Figure 3: Functionnal aspects of the architecture.

The three layers of the Control Unit are described now. Percepts gathered by receptors are the only information flowing bottom-up. The behavioural layer sends an atomic action to the motion module, basically an elementary action that could be executed by effectors. The planning layer produces a plan, ie a list of atomic actions. The cooperation layers may provide several plans. For each iteration, percepts are picked from the environment by the receptors and an atomic action is returned to the effectors. If there is a plan to accomplish, it will be realized as a sequence of atomic actions or behavioural procedures, until either the entire plan has been completed or current percepts or behavioural state indicates the plan is no longer valid. In the latter case, a new answer to current percepts will be computed climbing the decision layers.

**Conclusion and future work**

In this paper, the design of a reactive/cognitive architecture for emergency decision making has been described. A new contribution has been proposed by uniting the various decision making approaches for egress simulation in a flexible agent architecture for emergency situation. A functional overview of the architecture is purported in order to illustrate psychological and social adaptability provided by a layered design. At the moment, a partial implementation of this work for building evacuation has been realised in C++. The simulation platform is based on LGPL licence libraries such as OpenSceneGraph\(^1\) for graphic rendering and Bullet\(^2\) for physical collisions management. At the planification layer, the only managed goal is to get out of the building. The path planning process is based on a AStar implementation. Future works will focused on the goal selection model that must handle the case where multiple goals are possible.

**REFERENCES**


---

\(^1\)http://www.openscenegraph.org/projects/osg

\(^2\)http://www.bulletphysics.com/Bullet/wordpress
FITTING OF SEGMENTED GAUSSIAN PLUME MODEL PREDICTIONS ON MEASURED DATA

Petr Pecha and Radek Hofman
Institute of Information Theory and Automation
Pod Vodarenskou vezi 4
182 08 Prague 8, Czech Republic
E-mail: pecha@utia.cas.cz

KEYWORDS
Pollutant Spreading, Simulation Models, Uncertainty Bound

ABSTRACT
An improvement of mathematical model predictions of environmental pollution can be achieved on basis of assimilation of model simulations with real observations incoming from terrain. In this article we pay attention to development and investigation of applicability of one simple empirical method of objective analysis based on least square approach. Output background fields of resulting potentially dangerous endpoints are modified by measurements in such a way, that resulting respond surface is fitted towards measurements through the iterative adjustment of a certain selected set of model input parameters. In spite of a certain limitations this approach has occurred to be applicable for the first preprocessing of the model predictions and simulated measurements. It can support robustness of decision making and can contribute to early detection of possible fatal decision maker errors due to misinterpretation of input parameters of an accidental release scenario.

ASSESSMENT OF ACCIDENT CONSEQUENCES
Potential failures occurred in man-made processes can cause dangerous phenomena resulted in accidental releases of harmful substances into the living environment. Hazard evaluation and decision-making focused on early warning and protection of population has the highest priority. Reliable and up to date information represents basic inevitable conditions for effective management of intervention operations targeted on consequence mitigation during emergency situations. This appeared to be a basic lesson for further progress of emergency preparedness procedures, which has arisen from Chernobyl accident where lack of reliable information has shown to be the main reason of poor effectiveness of countermeasures. Decision making has to be supported by proper user-friendly simulation software tool complied with advanced theoretical methodology with access to all necessary relevant latest data. Crisis management should come out from reliable simulation of space and time of accident evolution, which should take into account all available information including physical knowledge of problem, expert judgement of input data, online measurements from terrain and others. The subject of investigation concerns evaluation of consequences of radioactivity propagation after an accidental release from nuclear facility. Transport of radioactivity is simulated by mathematical models from initial atmospheric propagation, deposition of radionuclides on the ground and spreading through food chains towards human body. In the final step a hazard estimation based on doses of irradiation is integrated into the software system HARP. Our access is mentioned in (Pecha et al. 2007).

FROM DETERMINISTIC TO PROBABILISTIC APPROACH AND DATA ASSIMILATION
Recent trends in risk assessment methodology insist in transition from deterministic procedures to probabilistic approach which enables generate more informative probabilistic answers on assessment questions. Corresponding analysis should involve uncertainties due to stochastic character of input data, insufficient description of real physical processes by parametrisation, incomplete knowledge of submodel parameters, uncertain release scenario, simplifications in computational procedure etc. Simulation of uncertainty propagation through the model brings data not only for the probabilistic assessment mentioned above (Pecha et al. 2005) but also for another main task of analysis called assimilation of model predictions with real measurements incoming from terrain. Data assimilation represents the way from model to reality and can substantially improve the model predictions. There are several important sources of information that can enter the assimilation procedures. Basic physical knowledge is included in prior fields (resulted vectors) predicted by simulation model. Assumptions related to the random characteristics of model inputs are supported by some kind of expert judgements (Goosens 2001). Substantial benefit can result from accessibility of data incoming from terrain. Merging of all these contending resources is a principle of assimilation and had shown to be very promising in many branches of contemporary Earth sciences (e.g. Drécourt 2004). Each such resource can be known on a certain degree of details (e.g. dense or rare measurements in space and time, complete or only partial knowledge of model error covariance structure, cases with indirect observations etc). Available information determines the option of suitable assimilation technique. We are considering the assimilation techniques in broader sense (Hofman 2007) from simple interpolation (poor model predictions, but dense and precise observed data) up to advanced statistical methods when full description of error covariance structure is needed - e.g. in (Kalnay 2003).
DATA ASSIMILATION (DA) USING MINIMISATION TECHNIQUE (MT)

In this article we are introducing one simple particular method based on nonlinear optimisation technique. During assimilation we assume precise measurements and thus the procedure cannot be presented as pure statistical DA. On the other hand it requires proper environmental model which is able to describe uncertainty propagation (Pecha et al. 2005). Our model is based on segmented Gaussian plume model (SGPM) approach that can account approximately for dynamics of released discharges and short-term forecast of hourly changes of meteorological conditions. For near area from the source and constant meteorological conditions can be used also simplified version of Gaussian straight-line plume model (GPM). Implemented numerical difference scheme enables simulate approximately formation of important parent-daughter pairs.

The objective multi-dimensional function $F$ of $N$ variables (subjected to bounds) is minimised starting at initial estimate. Commonly used Nelder-Mead direct search or Powell minimisation methods are tested here for elementary scenarios of accidental harmful discharges. Applicability bounds are examined for which satisfactory results at acceptable time of computation are achieved.

PRINCIPLES OF APPLICATION WITHIN ATMOSPHERIC DISPERSION MODELLING

Even for the simplest formulation of atmospheric dispersion and deposition in terms of Gaussian straight-line propagation the model $M$ is nonlinear. In the following paragraphs we shall concentrate on accidental radioactivity release into atmosphere and its further deposition on terrain. Approximation in terms of source depletion scheme accounts for removal mechanisms of admixtures from the plume due to radioactive decay and dry and wet deposition on terrain (Pecha et al. 2007). Let us proceed directly to the examination of the resulting fields of radioactivity deposition of a certain nuclide on terrain. The output is assumed to be represented by vector $Z$ having dimension equal to the number $N$ of total calculating points in the polar grid (in our case $N = 2800$, what means 80 radial sections and 35 concentric radial zones up to 100 km from the source of pollution). General expression for dependency of $Z$ on model input parameters $\Theta_1, \Theta_2, \ldots, \Theta_K$ can be formally written as

$$Z = M(\Theta_1, \Theta_2, \ldots, \Theta_K) \quad (1)$$

Let there are $R$ receptor points on terrain where the respective values are measured. Generally, the number of receptors is much lower then $N$ and we meet the problem with rare measurements expressed by observation vector $Y = (y_1, y_2, \ldots, y_R)$. Positions of sensors generally differ from the points of calculation grid. We shall use terminology from data assimilation for introduction of observation operator $H$, specially for its linear observation matrix $H$. $H$ is $R \times N$ matrix and transforms vectors $Z$ from model space (having length $N$) into corresponding vector $\tilde{Z}$ in observation space (having length $R$) according to matrix notation $\tilde{Z} = H \cdot Z$. Components $\tilde{z}_r$ of vector $\tilde{Z}$ represent model predictions interpolated at the positions of simulated observations $r = 1, \ldots, R$. Innovation vector $D = Y - H \cdot Z$ is defined.

![Induced Deformations](image)

Figure 1: “Manipulations” with Resulting Straight-line Gaussian Shape

Number $K$ of input parameters is rather high (several tenth) and then for practical purposes only $S$ of them are treated as random. Rest of them are assumed to be less important from viewpoint of uncertainty propagation through the model and we assign them their best estimated values. Equation (1) is then simplified to the form

$$Z = M(\Theta_1, \Theta_2, \ldots, \Theta_S, \Theta_{S+1}, \ldots, \Theta_K) \quad (1)$$

In other words a certain number $S$ of selected problem-dependent optimisation parameters $\Theta_1, \Theta_2, \ldots, \Theta_S$ are considered to be uncertain and subjected to fluctuations within some range. The function $F$ is constructed as a sum of squares in the measurement points between the values of model predictions and values observed in terrain:

$$F(\Theta_1, \ldots, \Theta_S) = \sum_{r=1}^{R} (y_r - \tilde{z}_r(\Theta_1, \ldots, \Theta_S))^2 \quad (2)$$

Minimisation algorithm searches a minimum of scalar function $F$ of $S$ parameters starting at an initial “best estimate”. In brief glance, the test points $[\Theta_1, \Theta_2, \ldots, \Theta_S]$ of the objective function $F$ are arranged as a $S$-dimensional simplex and the algorithm tries to replace iteratively individual points with aim to shrink the simplex towards the best points. Model predictions can be interpreted as Gaussian surface (or superposition of partial Gaussian extents) over the terrain. Our objective is to take into account both model predictions and available measurements incoming from the terrain and to improve spatial distribution of deposited radioactivity. We can imagine the iterative process of minimisation of function $F$ such consecutive adjustment of the resulting respond surface, always according to the new evaluation of the parameters $[\Theta_1, \Theta_2, \ldots, \Theta_S]$. Thus, the predicted respond surface of results is gradually “deformed by permissible manipulations” directly driven by changes of problem-dependent optimisation parameters $\Theta$. MT algorithm controls the procedure until the best fit of modified surface with observation values is reached. Important feature of the method insists in preservation of physical knowledge, because the new set of parameters $[\Theta_1, \Theta_2, \ldots, \Theta_S]$ evaluated by minimisation algorithm always re-
enters the entire nonlinear environmental model \( M \) according to Equation (1).

**PRACTICAL IMPLEMENTATION AND RESULTS**

Investigation of applicability of minimisation assimilation technique was tested on so called “twin experiment”. Lack of real observations is simulated by artificial generation of measurements. Moreover, if we use for this generation the same environmental model (e.g. for a fix one set of disturbed input parameters) we can examine the problem convergence issues. In application part of the paper the results of two simulation experiments XWIN1 and XWIN2 are illustrated. XWIN1 relates to release of nuclide \(^{131}\text{I}\). Its further straight-line propagation and deposition on terrain is simulated according to simple scheme of straight-line Gaussian plume model. XWIN2 experiment deals with the problem of evolution of \(^{137}\text{Cs}\) deposition on terrain during the plume phase. Minimisation search is applied with more complicated but more realistic segmented model SGPM.

**MT applied to simple Gaussian straight-line model**

Accidental one-hour release of radionuclide \(^{131}\text{I}\) with total radioactivity 1.28 E\(^{+11}\) Bq discharged into atmosphere from nuclear facility is analysed. Release height is 100 m, propagation continues under constant meteorological conditions (straight-line propagation in direction North-East, mean plume velocity 1.6 m.s\(^{-1}\), Pasquill category D of atmospheric stability, no rain). Atmospheric dispersion coefficients are calculated according to KFK-Jülich semi-empirical formulas.

In the first step all input parameters are assumed to be represented by their best estimate values denoted by \( \hat{0} \) and then the corresponding output vector \( Z^b \) presents deterministic solution of deposited activity of selected nuclide on terrain. At the same time \( Z^b \) represents initial estimate for starting of minimization iterative search. In the second step we shall further reduce the number of parameters S from equation (2) to four parameters. Corresponding four uncertainties \( c_1, c_2, c_3, c_4 \) are introduced into the model according to scheme \( \theta_i = c_i \cdot \hat{0}_i \) or \( \theta_i = \hat{0}_i + c_i \cdot f(\hat{0}_i) \). Specifically, their meaning, usage in the environmental code and real choice is given in Table 1.

The function \( F(\theta_1, \theta_2, ... , \theta_S) \) from (2) now has form \( F(c_1, c_2, c_3, c_4) \) and minimization algorithm handles with 4-dimensional simplex. For purposes of construction of function \( F \) we have used slight modification of probabilistic version of existing environmental model HARP (Pecha et al. 2007) where original random inputs \( c_1, c_2, c_3, c_4 \) now play more general role of uncertainties characterized only by their range of possible fluctuations (see column 4 in Table 1). Minimisation algorithm uses this constraints such lower and upper bounds for permissible manipulations with values of variables \( c_1, c_2, c_3, c_4 \) (see arrows in Figure 1). During XWIN experiments the observation vector \( Y = (y_1, y_2, ..., y_n) \) is simulated artificially, the simplest way is utilization of the same environmental model \( M \).

<table>
<thead>
<tr>
<th>parameter</th>
<th>Unit</th>
<th>uncertainty inside code</th>
<th>uncertainty bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_1 ): Source release rate</td>
<td>[Bq.s(^{-1})]</td>
<td>( Q = c_1 \cdot \hat{Q} )</td>
<td>( c_1 \in &lt;0.1,2.9&gt; )</td>
</tr>
<tr>
<td>( \theta_2 ): Horizontal dispersion</td>
<td>[m]</td>
<td>( \sigma_x = c_2 \cdot \hat{\sigma_x} )</td>
<td>( c_2 \in &lt;0.1,3.1&gt; )</td>
</tr>
<tr>
<td>( \theta_3 ): Wind direction</td>
<td>[rad]</td>
<td>( \varphi = \hat{\varphi} + \Delta \varphi )</td>
<td>( \Delta \varphi = c_3 \cdot \frac{\pi}{80} )</td>
</tr>
<tr>
<td>( \theta_4 ): Dry depo. velocity</td>
<td>[m.s(^{-1})]</td>
<td>( \nu = c_4 \cdot \nu_0 )</td>
<td>( c_4 \in &lt;0.1,4.0&gt; )</td>
</tr>
</tbody>
</table>

Deterministic best estimate distribution \( Z^o \) generated on the polar calculation grid in original wind direction \( S_{\text{orig}} \) (North-East) is drawn in figure 2 as TRACE 1. It corresponds to the best estimate values \( \{ c_1, c_2, c_3, c_4 \}_{\text{best}} = \{ 1.0, 1.0, 0.0, 1.0 \} \). Selected positions of observations are labelled by red circles. For simulation of measurements in this red points we have selected a certain fixed quartet \( \{ c_1, c_2, c_3, c_4 \}_{\text{obs}} = \{ 1.73, 1.51, +4.00, 1.98 \} \). Artificially simulated measurements were generated using vector \( Z_{\text{obs}} = M \{ c_1, c_2, c_3, c_4 \}_{\text{obs}} \). Then the values are transformed into observation positions according to \( Z_{\text{obs}} = H \cdot Z^o \). Final simulated observation vector is obtained by assignment \( Y = Z_{\text{obs}} \).

![Figure 2: \(^{131}\text{I}\) Deposition Levels [Bq.m\(^{-2}\)] Related to the End of Plume Progression. XWIN I experiment using Gaussian straight-line model. TRACE 1 and TRACE II are initial best estimate and resulting assimilation with simulated measurements (at red circles)](image)

Minimisation algorithm in successive iterations \( j \) brings newly generated quartets \( \{ c_1, c_2, c_3, c_4 \}_{\text{close}} \) closer and closer to the \( \{ c_1, c_2, c_3, c_4 \}_{\text{obs}} \). Fast convergence of assimilated model predictions towards simulated observations has been found. 220 iterations are calculated during about 6 minutes and the following values has been found: \( \{ c_1, c_2, c_3, c_4 \}_{\text{close}} = \{ 0.731, 1.514, +4.003, 1.982 \} \). It demonstrates very good consent with “simulated” observations generated by \( \{ c_1, c_2, c_3, c_4 \}_{\text{obs}} \). The results are illustrated in Figure 2 as TRACE II isolines.

Original best estimate deposition on terrain (and at the same time initial guess entering MT) is labelled as TRACE I. Deposition after 220 iterations is calculated as \( Z_{\text{close}} = M \{ c_1, c_2, c_3, c_4 \}_{\text{close}} \) and its isolines illustrates TRACE II. The assimilated respond surface TRACE II is at the same time practically identical with
$Z^{obs}$ generated according to $M$ ( $\{ c_1, c_2, c_3, c_4 \}$) originally used for artificial simulations of measurements. The shapes of TRACE I and TRACE II reflect imposed changes in values of $c_1^{best}$ to $c_1^{obs}$ (higher nuclide discharge), $c_2^{best}$ to $c_2^{obs}$ (higher peripheral dispersion), $c_3^{best}$ to $c_3^{obs}$ (twist by 18°), $c_4^{best}$ to $c_4^{obs}$ (more intensive dry deposition causing steeper longitudinal gradient).

Direct search algorithm connected with Gaussian straightline propagation model has proved fast convergence provided that the measurements are well positioned. Its applicability depends on validity and limitations of model itself (more e.g. in (Irving 2004) ). However, the TWIN I results support an idea of MT application for preliminary fleeting estimation in near distances and during constant meteorological conditions.

**MT with more realistic SGPM environmental model**

TWIN2 scenario is formulated in connection with segmented Gaussian plume scheme (model SGPM marked as $M_{SGPM}$), which is much more complicated than straight-line spreading (our approach described in (Pechá et al. 2007) ). The model synchronizes segmentation of release dynamics with hourly meteorological forecasts. The first two consecutive release segments of $^{137}$Cs discharge (each with 1 hour duration) with released amount 2.0 E+17 Bq and 1.0 E+17 Bq has character of severe LOCA accident with partial fuel cladding rupture and fuel melting. Short-term meteorological forecast for the next 48 hours is provided by the Czech meteorological service. Then, for each hour since the release initiation there are available predictions of wind direction and speed, category of atmospheric stability according to Pasquill classification and rain precipitation. Omitting other details, the TWIN II scenario covers period of the first 3 hours from the release start and we are declaring the following plan:

**i)** Each of the two segments is modelled up to third hour after the release start taking into account short-term hourly meteorological forecast. The situation just after 3 hours is given by superposition of both segments in their successive meteorological hourly phases (5 phases in total). Resulting best estimate fields are calculated in analogy with Equation (1) according to schematic $Z_{3hr}^{best} = M_{SGPM}$ ( $\{ c_1, c_2, c_3, c_4, c_{33}, c_{34}, c_{51} , c_{52}, c_{53} \}$) and is illustrated in Figure 3a as TRACE I.

**ii)** Number of uncertainties is increased from four to five as $c_{33}$, $c_{34}$, $c_{51}$, $c_{52}$, $c_{53}$ stands for fluctuation of mean wind velocity. If we suppose wind direction and velocity fluctuations to be independent between hourly phases, then $c_4$ and $c_5$ split to 6 independent uncertainties $c_{31}$, $c_{32}$, $c_{33}$ (for wind direction predicted for hours 1, 2, 3) and $c_{51}$, $c_{52}$, $c_{53}$ (for wind velocity predicted for hours 1, 2, 3).

**iii)** We have simulated artificially fictive “observation surface” according to $Z_{3hr}^{obs} = M_{SGPM}$ ( $\{ c_1, c_2, c_{31}, c_{32}, c_{33}, c_{34}, c_{51}, c_{52}, c_{53} \}$) $^{(b)}$. Vector of simulated measurements at observation positions (see black filled squares in Figure 3b) are calculated by help of linear observation operator as $Y_{3hr} = H Z_{3hr}^{obs}$. Their incoming is supposed at one stroke just at hour 3 after the accident start. Let us state beforehand that assimilated TRACE II from Fig. 3b nearly corresponds with the “observation surface”.

**iv)** The main goal is to accomplish assimilation of the model predictions $Z_{3hr}^{b}$ in compliance with measurements $Y_{3hr}^{obs}$ in analogy with equation (2) using BCPOL procedure of minimisation.

**Figure 3a**: Nominal Deposition of $^{137}$Cs (just 3 Hours after the Release Start)

Deposition of $^{137}$Cs on terrain after 728 iterations is calculated as $Z_{3hr}^{728} = M_{SGPM}$ ( $\{ c_1, c_2, c_{31}, c_{32}, c_{33}, c_{34}, c_{51}, c_{52}, c_{53} \}$) and its isolines illustrates a trail on terrain marked as TRACE II. The results represent a new distribution just at third hour after the release start, which is improved by observations. Minimisation algorithm is initiated by the best estimate solution (TRACE I) and gradually approaches to the simulated observations. In short numerical summary, TWIN2 experiment required to prepare in advance sets of parameters { $c_1, c_2, c_{31}, c_{32}, c_{33}, c_{34}, c_{51}, c_{52}, c_{53}$ } for:

*best estimate:* $\{ ... \}^{b} = 1.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0$ 

*measurements:* $\{ ... \}^{obs} = 7.0, 2.0, 4.0, 5.0, 4.0, 2.5, 0.5, 0.6, 0.7$

Here are examples for a particular iteration $j$:

$\{ ... \}^{j=728} = 7.18, 2.49, -3.94, -5.80, -6.34, 2.49, 0.21, -0.28, -0.59$ (*values*)

$\{ ... \}^{j=1200} = 7.25, 2.03, -4.14, -5.80, -6.39, 2.59, 0.27, -0.36, -0.58$

(*) TRACE I in Figure 3a, (**) close to TRACE II in Figure 3b, (***) TRACE II in Figure 3b

Meaning of the parameters $c_1$ to $c_4$ is the same as described in Table 1. $c_5$ stands for uncertainty of the mean velocity of the plume. Further splitting to $c_{51}$, $i=1,2,3$ holds true for independent fluctuations of the mean velocity $\overline{\dot{u}}$ forecasted for hours $i$. Uncertain $\overline{\dot{u}}$ is then expressed according to $\overline{\dot{u}} = \overline{\dot{u}}^{best}$ (1+0.35$c_5$). $c_5$ bounds are $<-1, +1>$. More detailed recommendations for uncertainty bounds arising from expert judgement can be found e.g. in (Goossens at al. 2001).
TWIN II experiment took into consideration 9 optimisation parameters with constructive idea to discriminate according to their global or local effect (introduced into the wind vector). The computation procedure is time consuming, but satisfactory convergence can be achieved. The question of real field measurements (e.g. Eleveld 2004) is so far opened.

Figure 3b: Assimilation of Predicted Deposition of $^{137}$Cs with Measurements just 3 Hours after the Release Start (artificially simulated measurements in black squares)

CONCLUSION

Advantage of utilisation of SGPM output fields as a fitting surface insists in preservation of physical knowledge of the model. Presented experience related to applicability of minimisation techniques indicates that number of selected optimisation parameters c should not be too high in order to avoid the poor convergence or even taking the wrong way (more sophisticated algorithms have to be tested). At this stage we recommend to consider five optimisation parameters included in the TWIN II experiment (where wind velocity vector is global, it means no further splitting of c3 to further c3i and c5 to c5i) and link the 6th parameter c6 representing uncertainty in precipitation intensity forecast.

Presented minimisation technique fits the model simulation results on a certain specific situation. Any resulting effect (e.g. peripheral plume dispersion) usually depends on many other input random parameters. Thus, in no case the presented fitting technique should not be confused with parameter calibration. The problem of handling of real measurements still remains opened, the first considerations for the Czech territory are discussed in (Kuca et al. 2008). Presented approach can play a specific role among empirical assimilation techniques, especially as fast and efficient software tool for analysis of possible discrepancies between the model predictions and observations incoming from terrain. The method is incorporated into assimilation subsystem the HARP code (Hofman et al. 2007).

Realistic prediction of evolution of radiation situation during emergency gives decision makers necessary time on judgement and introduction of efficient urgent countermeasures on population protection. Reliable model predictions for the next hours in medium distances should account both for implementation of spatial meteorological forecast and development of new numerical techniques for time update of the trajectory models (e.g. how to propagate model for the next hours starting from assimilated results TRACE II in Figure 3b). Interventions introduced on the basis of non-assimilated TRACE I could lead to fatal consequences on population health resulting from ill-anticipated impacted areas.

ACKNOWLEDGEMENTS: This work is part of the grant project GAČR No. 102/07/1596, which is founded by Grant Agency of the Czech Republic. A lot of useful knowledge has been also acquired during RODOS customisation procedure for the Czech territory.

REFERENCES


SIMULATION MODELLING AND DATA-MANAGEMENT IN REAL-TIME SYSTEMS
RTDS: A COMPONENT AND ASPECT-BASED REAL-TIME DATABASE SYSTEM SIMULATOR

Leila Baccouche
Sami Limam
RIADI-GDL laboratory
INSAT, National Institute of Applied Science and Technology
C.U. Nord, B.P 676, Tunis Cedex 1080, Tunisia
E-mail: leila.baccouche@free.fr, limamsami@yahoo.fr

KEYWORDS
Simulation, real-time databases, components and aspects.

ABSTRACT

The demand for real-time data services has increased lately. Applications like stock markets, embedded systems and telecommunications are becoming more demanding real-time database management systems (RTDBMS). Research in the area of real-time databases leads to the development of sophisticated control algorithms and components and for testing these in different configurations we need a simulator framework that allows us to see how they behave for different transaction workloads. As a means for evaluating real-time database algorithms, we have developed RTDS, Real-Time Database Simulator, a simulator which is based on components and aspects. RTDS is fully configurable and has the ability of simulating a real-time database with differentiated services and distinct configurations. Using well-defined components, an application can use a RTDBMS with only the required functionalities starting with an architecture based on very simple admission control to quality of service support based on feedback control.

INTRODUCTION

Real-time systems and database systems are research areas that have been actively studied in the last decades. Due to the development of such systems, we observed an increase of applications manipulating great amount of data requiring specific RTDBMS. RTDBMS are specifically designed to fulfill both data requirements and real-time constraints. Although there is a variety of different database systems on the market, there is still a lack of real-time database systems. On the other hand there are projects that are developed by research teams and consist in real-time database prototypes.

The systems listed below represent some of the more recent real-time database platforms developed: STRIP (STanford Real-time Information Processor) (Adelberg et al. 1996), ARTS-RTDB (Kim et al. 1994), DeeDS (Distributed active, real-time Database System) (Andler et al. 1996), RODAIN (Taina and Raatikainen 1996), BeeHive (Stankovic et al. 1997), J-RADEX (Haubert et al. 2004) a Java translation of the RADEX RTDBMS simulator (Sivasankaran et al. 1994) improved by a new user interface and specific protocols. These databases are research project and are not component-based. They mainly consist of the same subsystems, e.g., interfaces, index system, and concurrency control system except for BeeHive which includes a real-time interface, a quality of service interface, a fault-tolerant interface, and a security interface.

Application developers must choose carefully the RTDBMS their application requires. A generic database platform that can be tailored such that it is suitable for different applications becomes an interesting solution. The component-based software engineering (CBSE) paradigm and flexible systems have emerged due to problems in traditional software development, such as high development costs and inadequate system evolution (Bosch 2000).

The CBSE paradigm has been successfully applied in many real-time environments. In database systems, components defined as new DBMS parts can be developed and integrated into a DBMS architecture which is no longer fixed. However to the best of our knowledge there are few component-based real-time database systems or simulators.

COMET is a software engineering approach for generating RTDBMS configurations suitable for resource-constrained automotive control systems (Nystrom et al. 2004). KIDS (Kernel-based Implementation of Database management Systems) has an architecture which consists of brokers and services (Geppert et al. 1997). Each service is provided by at least one component in the system and represents a specific task to be provided by the DBMS. Components in KIDS are DBMS subsystems that are collections of brokers. Brokers are responsible for a related set of tasks, e.g., object management, transaction management.

In this paper, we apply the emerging component developing paradigm and present RTDS a real-time database simulator consisting of a pre-defined set of components that an application developer can assemble in a different way to offer support for various application functionalities.
The rest of this paper is organized as follows. In next section we present MOA (multi-class Overload Architecture) the real-time database architecture implemented by RTDS. We then present the RTDS components and aspects platform. We describe the simulator environment and give an example of a real-time database scheduling algorithm simulated using RTDS. Finally we conclude the paper.

**MOA ARCHITECTURE**

MOA is a real-time layer with suitable real-time protocols allowing concurrent execution of multi-class transactions, support of various dispatching algorithms adapted to a multi-class transaction model and overload situations detection and resolution. All the algorithms are time-cognizant which is known to be a fundamental criteria in building RTDBMS. MOA supports main transaction models in the literature. Main models are: the model of Kang (Kang et al. 2002) (the transaction is not decomposed : it is an entity), the Milestone model (Liu et al. 1994) (the transaction is decomposed into a mandatory sub-transaction and several optional sub-transactions) and the (m,k)-firm model (Hamdaoui et al. 1995) (the transaction is decomposed into k sub-transactions: m are mandatory and the remaining sub-transactions are optional). MOA also supports single and multiple transaction queues and the associated scheduling algorithms.

MOA (Baccouche 2006) is composed of a transaction scheduler and a transaction manager. The transaction scheduler (TS) provides scheduling and dispatching algorithms which are specially designed for the multi-class transaction model. MOA’s last version includes sophisticated quality of service support based on feedback control. Figure 1 introduces MOA architecture:

![Real-time control layer](image)

**Figure 1: The MOA Architecture**

1. Admitted transactions are placed in an accepted queue. The transaction scheduler (TS) provides inserting and dispatching algorithms which are specially designed for the multi-class transaction model. Inserting algorithms are the most famous dynamic real-time scheduling algorithms: EDF and LLF (Liu and Layland 1973). The developer can use attributes like the importance to distinguish between transactions. Depending on their importance transactions are inserted in the appropriate queue. Most RTDBMS apply scheduling algorithms like EDF. In MOA, we use sophisticated dispatching algorithms which consider additional aspects in relation with the multi-class transaction model. Several algorithms of extraction are available: EDF, Background Scheduling, DBP an (m,k)-firm scheduling algorithm (Hamdaoui and Ramanathan 1995), DBP CC an (m,k)-firm scheduling algorithm combined with concurrency control check (Baccouche and Limam 2007). When a transaction execution is finished, the dispatching algorithm is executed in order to select a transaction for execution.

2. The transaction manager can be compared to a traditional database engine. MOA does not modify the internal functioning of the DBMS the code of which is difficult, it is well known, to obtain or to modify. Transactions extracted by the dispatching algorithm are inserted in a queue which is used by the transaction manager to execute transactions. It is responsible for transaction execution, consistency check, concurrency control and native transaction logging operations. The concurrency control algorithm applied by the transaction manager is 2PL-HP (Two phase lock with high priority) (Eswaran et al. 1976) which is free from priority inversion.

3. The quality of service manager is a set of controllers acting together in order to satisfy a given quality of service specification (in terms of data and transaction quality). It is part of a feedback control architecture (Kang et al. 2002). QoS guarantees is provided by using a performance monitor which continuously measures performance metrics. In addition the QoS manager compares measured values to the desired performance as given by the QoS specification and applies QoS algorithms to regulate system workload.

**THE COMPONENT AND ASPECT-BASED RTDS PLATFORM**

The RTDS simulator was motivated by the lack of free and open simulation environments. Its design was conducted to fulfill two main requirements. Our first goal is to take advantage of new software engineering advances. The second goal is to provide an online and free simulation platform.

The RTDS simulator has been designed to enable configurability so that it can handle a variety of different application requirements using different transaction models starting with flat models (Kang’s model (Kang et al. 2002)) to hierarchical transaction models with mandatory and optional parts. RTDS simulates a database system based on the MOA architecture and consists of a number of
components and aspects. Each component provides a well-defined service through operations that are defined in a component’s interface. Aspects provide optional functions that can be added to the system in order to improve some of its functionalities.

RTDS starts with a basic configuration that builds a fully functional RTDBMS capable of generating, scheduling and executing transactions. Basic components are (a) the transaction generator (b) the transaction scheduler (c) the queue handler which handles all transaction queues, (d) the transaction manager and (e) the RTDS core. Others components can be added to provide full system functionality and visualize results.

We also noticed in database component-based systems a lack in composition and deployment tools to assist the application developer in composing its system. Our goal through RTDS is to offer an online simulation platform to test a large variety of algorithms. Hence we developed a web interface to download and configure components installation.

RTDS Components

The component-based software engineering paradigm allows the treatments modularization and makes them reusable. Thus, it facilitates the application deployment and its update. For this purpose, we decomposed the simulator into components according to its functionalities. The following diagram shows the diagram of the simulator’s components:

![Simulator’s Components](image)

The 5 basic components necessary to the operation of the simulator are:

- **rtds.jar**: it is the component basic one from which the application will start; it contains the various mechanisms which make it possible to manage the components (activation, deactivation).
- **Scheduler.jar**: this component allows the insertion of a transaction in the suitable queue and its extraction for the execution. Insertion and extraction are done according to preset scheduling algorithms or algorithms newly defined by the developer (an editor allows to publish either an algorithm of insertion or an algorithm of extraction).
- **TransactionManager.jar**: this component allows the transactions execution while taking account of the resources conflicts by applying a concurrency control protocol.
- **QueueHandler.jar**: this component contains all the functions which allow the management of the queues.
- **TransactionGenerator.jar**: This component generates the transactions that are going to be executed by the simulator. The developer selects the transaction model to use and the number of queues where the generated transactions will be inserted. He has the choice between one, two or 3 queues. Once all choices are made, the transactions set is generated and stored in an xml file. Hence the same set can be reused for another simulation (using the transaction reuse optional component).

The other RTDS components are optional, which implies that we can deactivate them and leave them unused:

- **AdmissionController.jar**: This component has as a role to check the acceptance of incoming transactions under preset conditions.
- **Console.jar**: this component has as a role to post messages relating to the progression of simulation to hold the user informed of the state of its evolution.
- **Transactionreuse.jar**: As the result of transactions generation is an XML file, this last could be re-used for another simulation in order to compare the performance of different algorithms.
- **TransactionViewer.jar**: this component makes it possible to visualize the whole of the transactions generated in a table.
- **QoS.jar**: this component introduces quality of service support to RTDS. Thus it parameters and specifies the actions to be undertaken function of the system load in order to guarantee the QoS specification given by the application developer.
- **Help.jar**: this component illustrates how to use the simulator.

We have observed that the architecture of component-based systems vary from fairly fixed (extensible systems), to completely configurable. In the extensible systems, which have fixed architecture, extensions are allowed only in well-defined places of the architecture. RTDS is a configurable system, its components can be freely added or exchanged in the architecture. For example the transaction viewer can be exchanged with a new viewer component developed by the application user, displaying a tree of transactions or any other format. RTDS allows significant amount of flexibility to the developer and the user, in comparison to other systems. The QoS component can be included, in that case RTDS will simulate an autoregulated system based on the feedback control architecture. When discarded, the developer can control the system load using the admission controller component. Hence the database
can be tailored for new applications with different requirements. The developer can also edit his own scheduling algorithm using a defined pattern.

RTDS Aspects

There are functionalities of the database system that cannot be encapsulated in a component with well-defined interfaces as they crosscut the structure of the overall system, e.g., performance measuring, logging, debugging, synchronization, memory optimization. Aspect-oriented programming (AOP) (Kiczales et al. 1997) has emerged as a new principle for software development that provides an efficient way of modularizing crosscutting concerns by encapsulating them in special modules called aspects. Aspects are described separately and used through pointcuts which represent the point in the component code where the aspect is called. In our simulator, we defined two aspects:

- The monitor: the monitor makes it possible to measure the system performance and to represent them according to a graph which traces the rate of failure according to the system load.
- The logger: the logger keeps a trace, in an XML file, of the life cycle of all the transactions. Here’s examples of some traces: \( T_{12} \) has been inserted in the \( H\_\text{queue} \) at time 1235. \( T_{24} \) has been extracted. \( T_{24} \) missed its deadline at time 1500.

USING RTDS TO SIMULATE REAL-TIME DATABASE ALGORITHMS

RTDS Environment

RTDS is a discrete-event simulator written in Java, designed to simulate real-time database behaviours. It is specially interesting to test and compare all algorithms that can be found in a RTDBMS, e.g., scheduling algorithms whatever they use single or multiple queues, concurrency control algorithms whatever they are pessimistic, optimistic or mixed, quality of service related algorithms, etc. A simple and convivial interface (fig. 3) allows the user to specify all his choices and to introduce simulation parameters.

Simulation Steps

To conduct a simulation, we follow a number of steps that are:

1. Generate transactions or reuse transactions given in an xml file
   a. Choose transaction model
   b. Choose queue model
   c. Specify the simulation time
   d. Specify the arrival rate
   e. Specify transaction timing parameters
2. Algorithms configuration
   a. Choose inserting algorithm
   b. Choose dispatching algorithm

3. Choose concurrency control algorithm
4. If Qos is activated, Configure QoS specifications
5. Start simulation
6. View results

![RTDS Environment](image)

**Example of Performance Evaluation for the EDF Scheduling Algorithm**

For this purpose, we have generated five sets of transactions, one for each transaction arrival rate. The transaction arrival rate varies between 10 and 50 transactions per second, by increment of 10. The inserting and the extracting algorithms are EDF. The model of transactions is the Kang model and the model of queues is multiple queues where the number of queues is three. Finally, the parameters of transactions are the following: the load of periodic transactions is 0.4, execution time is between 30 and 70 ms for periodic transactions and between 30 and 150 ms for user transactions, the number of resources is 100 and each transaction can use from 1 to 5 resources.

![GMR for EDF, DBP and DBP_CC](image)

Fig.4 shows the global miss ratio GMR function of the arrival rate for 3 extraction algorithms: EDF, DBP and
DBP_CC. We note on this subject that it is possible to publish a graph which traces the miss ratio of each of the three queues (by class of transactions).

As can be observed, EDF performs a worse performance when the system is overloaded and the performance of DBP_CC is better than those of either DBP or EDF.

CONCLUSION

The use of simulators is sometimes a good way to prove algorithms’ performance and compare a set of algorithms. For this reason, we have developed RTDS : a Real-Time Database Simulator. By designing RTDS as a component and aspect-based simulator, we make it fully configurable and easy to tailor for different database requirements. RTDS is also upgradable indeed the developer can exchange a component with a newly developed one. Besides the developer can edit his own scheduling algorithm using the pattern given by the scheduler and following the steps described in the Help component. RTDS will be online in a near future. In this version of the simulator, we can’t simulate a distributed database but the associated transaction model is being studied so that this function will be included in a further version. We also plan to introduce design patterns in RTDS. For example in QoS management the pattern Observer can be used to detect system overload or underload and launch the associated actions to return to a steady state.

REFERENCES


Baccouche L. and Limam S., “(m,k)-firm scheduling of real-time transactions with concurrency control” in Proc of ICONS’07, April 2007, guadeloupe.


BIOGRAPHY

L. Baccouche received her Ph. D. in computer science from the National Polytechnic Institute of Grenoble in France in 1996. She’s an assistant professor at the National Institute of Applied Science and Technology in Tunisia in the computer science and mathematics department. Her research interest is related to real-time systems and real-time databases and includes scheduling, quality of service, feedback control and distributed decision.

S. Limam is a PhD student at the University of La Manouba in Tunisia. His thesis is about the management of transactions in imprecise real-time systems.
A FAULT TOLERANT REPLICATION CONTROL PROTOCOL FOR DISTRIBUTED REAL-TIME DATABASE SYSTEMS

Anis Haj Said, Bruno Sadeg and Laurent Amanton
University of Le Havre – LITIS Laboratory
25 rue Philippe Lebon, 76058 Le Havre Cedex, France
{anis.haj-said, bruno.sadeg, laurent.amanton}@univ-lehavre.fr

Bechir Ayeb
University of Monastir – PRINCE Laboratory
Boulevard de l’environnement, 5031 Monastir, Tunisia
bechir.ayeb@fsm.rnu.tn

KEYWORDS
Real-time database, transactions, distributed systems, replication protocol, fault tolerance.

ABSTRACT
Data replication is often used in distributed systems for both performance and fault tolerance purposes. In this paper, we are interested particularly to data replication in Distributed Real-Time Database Systems. We propose an optimistic replication control protocol which increases the chances of transactions to meet their deadlines and tolerates the loss of messages during the update process. We introduce a list, called List of Available Copies (LAC), associated with each data item in the database which contains a set of the most updated replicas identifiers. Fault tolerance is provided by building LACs dynamically according to transactions execution and system load, giving the real-time database a dynamic level of replication.

1. INTRODUCTION

In recent years, real-time database systems have been addressed in many research projects. Several papers have proposed many features in order to meet real-time transactions and data requirements. Some researches have addressed the transaction commit processing in distributed real-time database systems (DRTDBS). The main goal is to design distributed commit protocols which take into account the real-time transactions requirements. OPT (Gupta et al. 1996) and PROMPT (Harlisa et al. 2000) are two phase based commit protocols (2PC) (Gray 1978) proposed to this purpose.

Researches in distributed database area have addressed also data replication issue for both fault tolerance and performance purposes. Several replication control protocols are proposed to control replicas access and updates. In the literature, we distinguish three main models, namely Eager Replication (Gray and Reuter 1993; Bernstein et al. 1987), Lazy Replication (Gray and Reuter 1993; Pu and Left 1991) and On-Demand Replication (Adelberg et al. 1995). These replication models adopt different approaches to manage replicas. Eager Replication, uses a pessimistic concurrency control protocol to forbid the access to replicas of modified data and synchronizes all replicas before it commits. However, using this model can significantly increase the probability of deadlocks, and consequently failed transactions, notably when the transaction size and the number of nodes increase. In Lazy Replication, transactions are executed without worrying about the freshness of the handled data and the changes introduced by a transaction are propagated to other sites only after the transaction has committed. Eager and lazy replication control protocols are not suited to the real-time context because they don’t take into account the temporal constraints on data and transactions in real-time databases.

The On-Demand replication model is the most accommodated model to control real-time data replication. In this model updates are performed when a transaction needs an access to a replica. This model is often used in primary-copy architecture where the location of the most updated replica is static. A site sends an update request to the primary-copy in order to update data handled by a local transaction. This ensures the use of up-to-date data, but the time needed to update replicas can reduce the chances for the transaction to meet its deadline. Several On-Demand replication control protocols are proposed in the literature (Adelberg et al. 1995; Ahmed and Vrbsky 2000; Gustafsson and Hansson 2004; Wei et al. 2004). In (Wei et al. 2004) is presented ORDER (On-demand Real-time Decentralized Replication), a replication control protocol in which updates are performed periodically at a primary-copy site. Periods of updates are dynamically modified according to the needs of real-time transactions.

Another approach based on similar correctness criterion such as epsilon-serializability (Son and Zhang 1995) can also be fulfilled. Epsilon-serializability is based on the fact that in distributed real-time database systems, timeliness of results can be more important than their correctness (Bestavros 1996).

Most of the researches conducted on replication in DRTDBSs have addressed the control of real-time data replication. However, non real-time data are as important as real-time data because both are involved in real-time transactions, and then they are important for the system performance. A replication control protocol for DRTDBS must find a trade-off between ensuring replicas consistency and meeting transactions deadlines. This leads us to wonder about the level of replication of the database. Is fully replicated database the best approach? Is there a level of replication of the database that gives the best system performance? Is the level of replication should change according to the system overload? What replication model could allow to manage a partially replicated database? The answer to these questions is not given by the existing replication models.
The common drawback of Eager and Lazy Replication is that sites do not have any particular information about the state of the database at the other sites. In fact, for Lazy Replication, transactions are executed locally without carrying about the state of the copies of the involved data. Whereas for Eager Replication, transactions are forced to update all the copies of the involved data. The On-Demand model is generally used with a primary copy architecture in which the primary copy is assumed always up-to-date. In our work, we focus on replication of non-real-time data handled by real-time transactions in DRTDBS. We present DLR-ORECOP (Dynamic Level of Replication - Optimistic Replication Control Protocol), a replication control protocol which can be integrated into two-phase based real-time commit protocols. In fact, the execution of our protocol depends mainly on the commit protocol execution. The main contributions of DLR-ORECOP is propagating updates optimistically and providing a dynamic level of replication of the database. In addition, DLR-ORECOP tolerates different kinds of faults that may occur during its execution.

The remainder of this paper is organized as follows. Section 2 describes our models and assumptions. In section 3, we present DLR-ORECOP, a replication control protocol designed for DRTDBS. Section 4 shows how DLR-ORECOP allows the system to tolerate faults. Finally, Section 5 concludes the paper.

2. MODELS AND ASSUMPTIONS

The database is modelled as a collection of data items fully replicated at each node of the distributed system. The database may contain real-time data and non-real-time data. With a real-time data is associated a validity interval. A particular real-time data value is deemed useful only during its validity interval and gets out of date with the passage of time. A non-real-time data denotes a data whose value will not change with time, i.e. its validity interval is infinite.

For real-time transactions, we focus only on transactions with firm deadlines (Ramanritham et al. 2004). We distinguish two types of real-time transactions:

- Update transactions: generally, they are periodic sensor transactions and they update data which reflect the state of the environment. Update transactions should refresh real-time data before exceeding their validity intervals.
- User transactions: some transactions launched by users or triggered by some events must be performed before exceeding their deadline, otherwise the result will not have any significance for the system. User transactions deal with both real-time and non-real-time data, except writes on real-time data which are performed only by update transactions.

The distributed systems $S$ consists of a set of sites interconnected by high-speed network.

$$S = \{S_1, S_2, \ldots, S_n \mid n \geq 2, n \in \mathbb{N}\}$$

Where $n$ is the number of sites and $S_i(2 \leq i \leq n, n \in \mathbb{N})$ is the identifier of the $i$th site.

We assume that a global transaction may arrive at any site. The specification of global transactions is as follows:

$$GT = \{GT_{id}, MS, DS, CS, DL, SF, STS, STATE\}$$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT_{id}</td>
<td>global transaction identifier</td>
</tr>
<tr>
<td>MS</td>
<td>master site identifier (coordinator)</td>
</tr>
<tr>
<td>DS</td>
<td>data set</td>
</tr>
<tr>
<td>CS</td>
<td>set of cohorts identifiers</td>
</tr>
<tr>
<td>DL</td>
<td>deadline</td>
</tr>
<tr>
<td>SF</td>
<td>slack factor</td>
</tr>
<tr>
<td>STS</td>
<td>set of subtransactions identifiers</td>
</tr>
<tr>
<td>STATE</td>
<td>current state</td>
</tr>
</tbody>
</table>

Table 1. The global transaction specifications

The set of cohorts $CS$ contains $k$ sites identifiers ($2 \leq k \leq n, k \in \mathbb{N}$). The data set $DS$ consists of a collection of data objects involved by the global transaction. In each data object $DO_i$ involved by the global transaction are specified the following elements.

$$DO_i = \{D_{id}, Op, C, V al, LAC, Lock\}$$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_{id}</td>
<td>a data item identifier</td>
</tr>
<tr>
<td>Op</td>
<td>the operation on the data item</td>
</tr>
<tr>
<td>C</td>
<td>the cohort identifier</td>
</tr>
<tr>
<td>Val</td>
<td>data value</td>
</tr>
<tr>
<td>LAC</td>
<td>list of Available Copies</td>
</tr>
<tr>
<td>Lock</td>
<td>lock value</td>
</tr>
</tbody>
</table>

Table 2. Data Object Specification

User transactions are split into subtransactions which are executed at different nodes of the system according to locations of up-to-date data. In order to ensure transactions atomicity, the system uses commit protocols. Among commit protocols, we focus on two phase based real-time commit protocols, and particularly on PROMPT (Harita et al. 2000).

3. THE DLR-ORECOP PROTOCOL

The originality of our protocol consists of introducing a new entity, called List of Available Copies (LAC) which allows the database to have a dynamic level of replication. In fact, we store a LAC with each replica, alongside each data item it replicates. A LAC includes a subset of nodes that have the most updated state, according to the latest node that updated the data item. In the best case, a LAC includes all nodes identifiers of the distributed system. LACs are built dynamically according to the commit protocol processing and the transactions execution outcome. The optimism in our DLR-ORECOP is based on the assumption that most real-time transactions commit before their deadline. Based on this assumption, we allow each cohort to optimistically propagate its updates without waiting for the global decision of the coordinator.
The protocol is executed in four phases (Figure 1). The first phase consists of a distributed locking phase. It begins when a write operation on a non-real-time data item needs to be executed. The second phase begins after the completion of the subtransaction processing. It consists of propagating updates performed by the subtransaction and building new LACs. The cohort breaks off sending update messages from the beginning of the voting phase of the commit protocol. The third phase, consists of broadcasting new LACs, built at the previous phase, to the remainder sites of the system (except the coordinator). The latter phase is a validation phase (applying data and LACs updates). This phase is integrated in the decision phase of the commit protocol. In this phase, sites decide to validate or nullify updates received from cohorts.

In the following, we detail the different phases of DLRORECOP protocol.

### 3.1. The distributed locking phase

During transaction processing, the access to replicas of modified data must be avoided until their update. In fact, when a transaction needs to perform a write operation on a data item, it must at first hold a write lock on the local copy and on all its replicas. The protocol uses a distributed locking approach. Thus, if the lock is granted by all sites, then the transaction can proceed. If not, it can be delayed and the request is repeated some time afterwards. When a replica of a data item is locked for write access, its LAC is modified and contains only the identifier of the site in which the transaction that holds the write lock is executed. At this time, any incoming transaction at any site of the system which needs an access to this data is automatically launched on the site whose identifier is contained in the LAC related to that data item. In fact, when a transaction is launched at one site, it refers to the LACs of the needed data in order to locate the most recently updated replicas. Thus, a transaction is divided into subtransactions which are launched at sites according to up-to-date location.

### 3.2. Optimistic updates propagation

When a cohort sends the WORKDONE message to the coordinator, it waits for the beginning of the voting phase which should be initiated by the coordinator. The duration of the waiting time depends on the reception time of the last WORKDONE message by the coordinator. This time can be relatively long since it depends on sites overload, length of the execution times of subtransactions and messages delivery time. While waiting for the voting phase, we optimistically assume that the transaction is normally executed and will be successfully committed. Based on this assumption, we allow each cohort, which has successfully completed its subtransaction processing, to optimistically propagate updated data and waits for acknowledgements. The identifier of each sender of an acknowledgement message is added to the LAC associated with updated data items. Thus, a LAC built by the cohort contains sites identifiers in which replicas will be certainly updated if the transaction commits. A cohort stops propagating updates as soon as it receives a PREPARE message from the coordinator. Data updates and their associated LACs are joined to the vote message in order to update coordinator’s replicas.

### 3.3. LACs propagation

At this stage, the cohort has entered a prepared state wherein it cannot unilaterally commit or abort the transaction, but has to wait for the final master decision. When the coordinator initiates the vote phase, the transaction abort becomes unlikely since all cohorts have successfully performed their subtransactions and then survived to each possible concurrency conflict. While the transaction waits for the global decision, it broadcasts new LACs to the other sites. The main goal is to allow sites to make a suitable decision when an incoming transaction needs an access to an updated data.

### 3.4. Global decision propagation

The end of the prepared state takes place when the global decision sent by the coordinator is received by the cohort. The coordinator sends its decision to all sites of the distributed network. Two cases can occur:

- The coordinator decision is COMMIT: Here, cohorts commit their subtransactions in normal fashion. All sites validate updates previously received by subtransactions of that global transaction and release locks.
• The coordinator decision is **ABORT**: Here, cohorts abort their subtransactions in normal fashion. Updates propagated by these subtransactions are also nullified at all sites and locks are released. According to **DLR-ORECOP** protocol, some replicas do not receive updates yet. A replica is available if its site identifier is included in its **LAC**. Then, releasing locks do not affect the accurateness of transactions results. In fact, the access to stale data is indirectly forbidden by **LACs**.

4. FAULT TOLERANCE IN DLR-ORECOP

In this section, we show the usefulness of **LACs** in order to tolerate faults. The main drawback in using data replication is the heavy load generated when ensuring replicas consistency. If cohorts update each replica before committing a transaction, the probability to miss the transaction deadline can increase significantly.

4.1. Tolerating replica inconsistencies

As we have shown in section 3, the optimistic propagation phase is interrupted when the voting phase is initiated by the coordinator. The main purpose of the updates propagation interruption is to avoid that ensuring replicas consistency affects meeting transaction deadlines. Despite the inconsistencies that occur between replicas, the system performance is not affected since transactions use only up-to-date replicas. This is guaranteed by **LACs** which contain only site identifiers that have confirmed their replicas updating (by sending an acknowledgement for update message (**UM-ACK**)). Thus, when a site refers to **LACs**, it launches transactions at the suitable sites in which replicas are certainly up-to-date.

Figure 2 shows an example in which inconsistencies between replicas of a data item are allowed. After the cohort (site 2) has sent a **WORKDONE** message to its coordinator (site 3), **LACs** associated with replicas of the data item locked by the subtransaction executed at site 2 contains only the cohort identifier. The updates propagation phase is broken off by the voting phase initiated by the coordinator and only site 1 is added to the **LAC** associated with the updated data item. Thus, after the propagation of **LACs**, each site can locate up-to-date replicas of that data item despite the presence of inconsistencies.

4.2. Tolerating missing update messages

**LACs** are built dynamically by cohorts that hold write locks. The identifier of each site which has received an update message from a cohort, is not automatically added to **LACs** associated with the updated data. It is only done when an acknowledgement of the update message is received by the cohort. Otherwise, the site is considered as not updated yet due eventually to its crash or to an ordinary missing messages in the network or also to an overload state. Thus, cohorts are not forced to guarantee the update of all replicas in order to avoid the extension of transaction execution time.

![Figure 3. Missing Update Messages](image1)

Figure 3 shows an execution of **DLR-ORECOP** in a case of missing update message. Here, the update message addressed to site 4 is missed. This does not cause errors since the identifier of site 4 could not be added to the **LAC** of the updated data. Then, the replica of that data item at site 4 is not accessible, thus, no transactions will use it.

4.3. Tolerating missing LAC update messages

Here, after a cohort sends its vote, it propagates new **LACs**. If some messages are lost, there is no particular process to recover these messages. In fact, even if a site don’t receive a **LACs** updating message, its **LACs** still contain the lock holder identifier which is necessarily up-to-date. Thus, the access to an up-to-date replica is still possible. The process with which **LACs** are modified allows to have always at least one site in which replicas (associated with these **LACs**) is up-to-date. This gives to **DLR-ORECOP** the ability to conceal missing **LACs** updating messages.

![Figure 4. Missing LAC Update Messages](image2)

In Figure 4, site 4 don’t receives the **LACs** updating message. The **LAC** associated with its replica contains only
the cohort identifier in which the replica is certainly up-to-date. This information is sufficient for site 4 since it knows at least the location of one up-to-date replica.

4.4. Tolerating a prepared cohort crash

A crash of a cohort may occur during transaction process. The crash can be critical for the system when it arises after the transaction has sent its vote to the coordinator, i.e. during its prepared state. The crashing cohort tolerance is based on the reception of the global transaction outcome sent by the coordinator. In fact, the GT_{i,a} and the MS_{i,a} are specified in the lock request, then sites can validate or nullify updates when they receive the master decision. Indeed, in order to ensure transaction atomicity, prepared cohorts can not unilaterally commit or abort their subtransactions. Subtransactions outcomes are ultimately the same as the global transaction outcome which is decided by the coordinator. According to the global decision, one of the two following cases can arise. The coordinator decision is ABORT: If at least one cohort aborts its subtransaction, then the coordinator decides to abort the global transaction by sending an ABORT message to all sites. Then, cohorts abort their subtransactions by cancelling both data and their LACs modifications and by releasing locks. The remainder sites also cancel updates made by subtransactions and release locks on replicas The coordinator decision is COMMIT: In this case, the coordinator decides to commit the global transaction, it sends a COMMIT message to all sites. Then, cohorts commit their subtransactions and the remainder sites validate data and LACs updates. The crash is concealed since updated replicas become available. However, LACs could not be updated yet when the crash of a cohort occurs before the beginning of the LACs propagation phase. In this case, LACs contain only the crashed cohort identifier. Although some replicas are up-to-date, incoming transactions could not access them because sites are unable to locate them. This represents a drawback which affects the system reliability. In this situation, DLR-ORECOP ensures that at least one site receives LACs updates. This site is always the coordinator of the global transaction since LACs updates are joined to the vote message. Thus, before entering in critical phase, a cohort has already exported data and LACs updates. That is, these critical information can be propagated in the distributed system even when the cohort fails. A LAC could not be updated for two reasons. Either the message is missed, or the sender of the expected message is crashed. In both scenarios, a site can send a request for missed updates to the coordinator. However, this process is triggered only when the site detects the failure.

5 Conclusion

In this paper, we have proposed a novel replication control protocol designed for distributed real-time database systems. The DLR-ORECOP protocol integrates features in order to adapt data replication control to the real-time environment. With DLR-ORECOP, updating all replicas is not necessary. In fact, the replication level of the database is dynamic and depends on the system load and on the evolution of the transaction execution. So, updating replicas do not cause the extension of the transaction execution time and then do not affect the system performance. Fault tolerance is provided by LACs which are built so that they conceal replicas inconsistencies caused by missing messages or sites crash while transaction processing.

References


DATA MODEL AND REAL TIME IN SPATIAL APPLICATIONS: WHAT DATA AND WHEN?

Michel Mainguenaud
Institut National des Sciences Appliquées (Rouen) – LITIS Lab
Avenue de l’Université
F-76801 Saint Etienne du Rouvray, France
E-mail: michel.mainguenaud@insa-rouen.fr

KEYWORDS
Database model, Real time applications, Mobility, Data Definition Language

ABSTRACT
A real time application provides some time constraints in the delivering of database query results. We consider here the database model level to handle such constraints. We rely on the time sensitive object concept and we increase these notions with database properties such as extended functional dependencies, characterized aggregate functions and the definition of an order on a lattice. Depending on the available time to respond to a query, relevant data are provided in accordance to the perception of the data model defined for a given application. The extended functional dependencies involve a 1 to 1 correspondence between several entities of the data model. The order on the lattice defines the priorities. Characterized aggregate functions introduce the N to 1 correspondence and therefore require a more precise definition rather than a single trigger definition. The definition takes into account the relationships of the attributes involved in the aggregate function.

INTRODUCTION
The development of technologies, such as communication and positioning, increases real-time data traffic (Kennedy, 2002, Bossler, 2002). Intelligent Transportation Systems, Location Based Services require more and more Points of Interest within a navigation application. Therefore a database is required to store alphanumeric and spatial data. We define a geographic object as a set of couples (semantic property, value). An important point is to define the data to be provided as the response of a request. Real-time applications are based on the fact that whatever value an object has at any point in time, within a few milliseconds or seconds that value will cease to be valid within a brief time. Models for real-time systems traditionally focus on system processing and its timing characteristics. Timing constraints for real-time systems are typically expressed with respect to processing. In this article, we prefer considering geographic objects in the sense of time-sensitive objects as defined in (Callison, 1995). This model focuses primarily on the time intervals over which object values may ideally be used, rather than the timing of process execution to describe the evolution of a system. The validity intervals are associated with quality indices upon data. To be able to choose the relevant data within a time constraint, a conventional data model should be improved. We introduce the strong dependency of the time sensitive objects constraints, the extension of a conventional database (DB) notion, functional dependency, as an extended functional dependency and an order on a lattice to define a real time application DB schema. Using these extra-meta-data with the data model, a query resolution mechanism is able to define a priority in the delivering of the schema to a mobile application.

The first part presents the problems in the real time environment. The second part presents the add-on in the data model. The third part presents the propositions to take advantage of the new add-on in the data model. The last part presents the conclusion and future works.

PROBLEMS
At the opposite sides in the classification of time validity are single-interval transient and immutable objects. Single-interval transient objects are created, remain in the system for a short time and disappear without changing. At the other extreme, are immutable objects, which persist for the life of the system execution without changing. Most objects in real-time applications fall between these two opposite sides. They persist for some finite time and may change during this lifetime. New values with associated validity intervals are defined (i.e., multi-interval object). We focus on the associated semantic alphanumeric data model to handle a real time application with a DB.

Managing a real time application with a DB is an old requirement (Taina and Son, 1997). Several formats or data models can be used for navigation services such as for example ISO-GDF, SDAL (Shared Data Access Library – NavTech). We are not concerned in this article by the spatial representation of the route nor the data model associated with the representation of such as route (Liu et al. 2005). Several DB transaction models are provided to manage the serialization in a multi-user application to support or not the predictions (Bodlaender and Stok, 1998). Others are designed to support routing in heterogeneous networks (Sun et al. 2007). We are not concerned in this article by the system transaction model or the data propagation within the communication network. Nevertheless we rely on the performances of such systems.

To handle time in a spatial context, several propositions are available in the literature (Guting et al. 2006, Parent et al. 1999, Gregersen, 2006). These propositions are based on an extension of a conventional definition for a DB schema, i.e., a pair (attribute name, domain of validity). The data model is given as a collection of data types and operations, which can
be plugged as attribute types into any DBMS (Data Base Management System) data model. The idea is to represent the temporal development of spatial entities in certain data types such as moving point, the balloon model, spatial abstract data types. These models consider that they do not have any time constraints during the query resolution mechanism (obviously every one would like to have a short response time of the system that implements the model, but every relevant data will be provided in the final results).

Our goal is to promote data definition properties to introduce an order of evaluation in a DB query resolution mechanism. In real time application, the DBMS may not have time to compute the complete answer of a query. Nevertheless, at least part of the answer could be provided as a result. The problem is to define which part should be provided first. Starting with a conventional definition of a DB schema based on a pair (attribute name, domain of validity), we must increase this definition with several notions since choices should be performed by the DBMS. The architecture is close to Dynamic Data Driven Real Time Management. We start with an Abstract Model as defined in (Forlizzi et al. 2000). The idea is to represent the temporal development of spatial entities in certain data types such as moving point or moving region. Suitable operations are provided on these types to support querying. Such data types can be embedded as attribute types into object-relational or other data models. It allows one to focus on the essential concepts and not get bogged down by representation details.

In the following, we consider a DB schema defined as a set of pairs (attribute name, domain of validity). The domain may be as complex as we like, e.g., a spatial representation with or without time management labels. The only requirement is that an attribute with a time management label must be documented, i.e., we do not accept a null value for the time management label when an attribute is defined as time-constrained.

MODELING TOOLS

To achieve our goals, we introduce in the DB schema several concepts to help the query resolution engine to define the relevant data to evaluate first: extended functional dependency, strong dependency, the characterized aggregate functions and the lattice.

Extended Functional Dependency

A functional dependency, as defined in conventional relational DBs (Ullman, 1995), indicates an assertion on the real world based on a 1 to 1 relationship.

Definition: Given a relation R of a DB schema, an attribute (or a minimal set of attributes) X in R is said to functionally determine another attribute (or set of attributes) Y, also in R, (written $X \rightarrow Y$) iff each X value is associated with precisely one Y value.

In this case, cycles are allowed. This definition is straightforward in the sense that X and Y must belong to the same relation (e.g., in the same class in UML for example if we want to translate this notion to the conceptual level to simplify).

We extend this definition by an Extended Functional Dependency in the sense that X is in the set of attributes of an entity E and Y may be an attribute (or a set of attribute) in another entity E' of the DB schema. This represents the extension of the 1 to 1 relationship in a conventional Entity-Relationship diagram or the 1-1 Association in a class diagram in UML.

The granule “sub-set” as defined in (Mainguenaud, 1994), applied with the inclusion operator provide the relevant candidates to an extended functional dependency.

Definition: An attribute is defined with a granule “sub-set” iff its semantic is valid for a sub set of its spatial representation. As an example, the value of the mayor attribute (i.e., the name of the mayor of a town) is a valid data for a sub-set of the spatial representation of a town (i.e., a mayor is also responsible for a sub part of the town). The value of the population is not valid for a sub-set of the spatial representation (i.e., no guaranty is provided that all the inhabitants of the town live in this specific sub-set of the spatial representation of the town).

The application of the inclusion operator means that a park is spatially included into a town. Therefore, an attribute classified as “sub-set” of a town is still valid for the park since the park represents a sub-set of the spatial representation of a town (since it is included into the town).

The graphical representation of such a pair of attributes is defined by a single circle on the attributes and an oriented edge as the edge in the definition. Figure 1 represents such an example (i.e., $<\text{idNumber}> \rightarrow <\text{Mayor}>$).

Strong dependency

The strong dependency, as defined in (Callison, 1995), indicates that the value of the dependent object depends critically on the value of the influencing object.

Definition: The value of the strong dependent object is to be updated as quickly as possible in response to any asynchronous change to the value of the influencing object.

To assure termination of the propagation process, cycles are prohibited in the strong-dependency relationship. We do not consider here, the update mechanism. The important point is that an update may occur (e.g., this update may be sporadic, periodic or any other way).

As an example, the speed of a car should be changed as soon as the car arrives in a new speed-limited area. In the DB model the attribute “carSpeed” should be updated as soon as “speedLimit” is changed (carCurrentSpeed $\rightarrow$ currentSpeedLimit). carCurrentSpeed is the strong dependent object and currentSpeedLimit is the influencing object.

The graphical representation of the strong dependency between such a pair of attributes (i.e., the strong one with a double circle and the influencing one with a single circle) is defined by an edge between these attributes (i.e., an oriented edge from the dependent one to the influencing one). Figure 1 represents such an example - i.e., $<\langle\text{carCurrentSpeed}\rangle \rightarrow <\text{currentSpeedLimit}>$.

Extended Functional Dependencies may accept cycles since they represent a semantic link (i.e., even without changes). Nevertheless, two attributes may be linked by an Extended Functional Dependency and by a strong dependency.
Characterized Aggregate Function

Aggregate functions may be useful to sum up a set of data and involve several levels of abstraction. An aggregate function can be provided with several critical precisions. The computational complexity of such a function is very important in a time-constraint environment. We define three levels of complexity: Linear (e.g., find the average value within a set of values), Quadratic (e.g., the number of tuples that satisfy a join operation in SQL), Super (i.e., higher than quadratic).

Managing navigation applications very often requires the definition of predictions (e.g., on the traffic, schedules). The aggregate functions may use a regression technique, an interpolation or any other types of models. The operations do not belong to the kernel of a DBMS. The application developer must provide an extra operator such as for example a trigger or an add-on operator in object-relational DBMS.

In a time constraint environment, the characteristic of an external evaluation of a function is important. The characterization of the function may be introduced with the Data Definition Language. As an example (Deshpande and Madden, 2006) provides a specific data definition order to create a view based on an interpolation technique (in this case for a set of sensors).

Lattice

Lattices offer a natural way to formalize and study the ordering of objects using a general concept known as the poset, partially ordered set (Buckley and Harary, 1990). A lattice as an algebra is equivalent to a lattice as a poset. Two elements of a lattice are incomparable if neither dominates the other.

We rely in this context on the ANSI/SPARC definition of a DB schema. The three levels are: the external level (i.e., views), the conceptual level (i.e., the global information system schema) and the physical representation (e.g., files, data organization or access with the file management primitives of the operating system). Only the external level is concerned by the proposed extensions. The conceptual level is global for an information system. It does not take into account the specificities of applications. Within the external level, we can take the whole semantics of an application. The data set is reduced (i.e., a sub set of the whole available schema in the conceptual level).

Depending on an application, the same entity may be considered as very important or as quite marginal. We take advantage of this difference of semantics to specify an order on the data model. We consider three levels: Essential attributes, Important attributes, Useful attributes.

Within a level the DB designer does not define an order. In this case, we have a partial order involved by the three levels. Let us consider the DB schema defined as a view on a global schema. Let us consider a mobile application using this schema (e.g., the maintenance of parks in a tourist region). Figure 2 presents the associated lattice.

As an example, a query is defined on such a schema and involves several classes. The aim is to help the DBMS to present relevant data depending on the time constraints. To simplify the presentation let us consider the following relational algebraic operators (Ullman, 1995): Π, the projection operator (i.e., the relevant attribute); σ, the selection operator (i.e., the conditions to be verified).

An example of a simplified query, Q, is defined on the toy DB schema using these operators:

\[ Q = \Pi (\sigma \text{Route.origin} = \text{‘ParkStartPoint’} \text{and destination} = \text{‘ParkArrivalPoint’}), \]

\[ \text{Park.name, carCurrentSpeed, Route.averageSpeed, Park.leisures}. \]

This query requires providing carCurrentSpeed that is involved in a strong dependency. It also can provide the responsible of the initial and arrival Park since mayor is involved in an Extended Functional Dependency with a park (via the idNumber of Park). It also requires the evaluation of a function with a small complexity (i.e., linear), the computation of the average speed (between the departure and the current time). The DBMS is then able to elaborate an execution plan with some specific priorities with respect to the order defined in the lattice. From the user-interface point of view, the presented data is not a unique query but in fact a set of queries. As we can see in Figure 1, the priority is given to the speed of the car versus other data. Complex data such as providing the current works in progress in the area (an aggregate function with a high complexity) would be provided if the computational time is available. One can remark that the lattice is query-independent. It is based on the application (i.e., the view).

Conclusion

Using these modeling tools, three levels are covered: static model with the extended functional dependency, dynamic on update with the strong dependency, and dynamic on resolution with the lattice and the characterized aggregate function.

To be able to evaluate the complexity of a function the overhead must be limited. In a time constraint environment, we must give the priority to a “compiled” approach. By “compiled” approach we mean that the maximum knowledge must be introduced into the data model or into the query. Managing a meta-base is simpler than evaluating information for each query.

Two main approaches may be defined: the first one is based on the query definition process; the second one is based on the DB schema. Both of them are compatible with the conventional view mechanism defined in DBMS. In both case, the query resolution mechanism must be changed since a transaction may be shortened to respect the time constraints. Therefore, the provided attributes (and the provided values) may be different depending on the context. The same query in different contexts does not provide the same results. This is the price to pay to provide an answer whatever the circumstances are.

With the first approach, the query definition process specifies the relevant relative importance of the required data. The query language must be extended to take into account this new requirement. The DB schema definition is still conventional.

| Essential = {Park.name, Car.carCurrentSpeed, PartRoute.speedLimit} |
| Useful = {Town.mayor, Route.averageSpeed} |
| Important = {Town.workInProgress} |

Figure 2 – Associated lattice
With the second approach, the query definition process is unchanged whatever the used device is (e.g., a mobile phone, a PDA or a micro-computer). The query language is unchanged. The DB schema definition is changed. The data model is increased with new concepts. The main advantage is that an application can be designed as a set of queries and the queries are device-independent (that does not mean that the presentation of the results are device independent). Therefore the current proposition is based on an extension of the DB schema.

**PRINCIPLES**

A conventional DBMS tries to provide a query result as soon as possible. No time constraint are involved. Whatever the conditions are, the same amount of data will be provided. In a real time constraint environment, the amount of data will be different since the length of a transaction may be different depending on the conditions). We propose to use some heuristics to enhance the accuracy of provided data. Tools defined in the previous section will be used to improve the accuracy of the result.

**Extended Functional Dependency**

A functional dependency is based on the fact that a value in the left part of the dependency provides a single value in the right part. One can notice that it does not mean that the right value is a single atomic value (e.g., it could be a set). This is not related with the notion of first normal form in relational DB. The sub-set granule property guarantees that the data is relevant in such a context.

**Definition H1:** In a time-constraint environment, we propose to give priority to attributes that are involved in an extended functional dependency.

It widely depends on the domain associated with the right value. A conventional data type (e.g., integer, real, string) allows taking full advantage of the situation. In a navigation application, providing part of a film, a song or a full text document that popularized the place will change the situation.

**Strong dependency**

A strong dependency will require that the left value of the dependence is updated as soon as possible. The quality of data is therefore enhanced.

**Definition H2:** In a time-constraint environment, we propose to give priority to attributes that are involved in a strong dependency.

We do not have the guarantee that the update has been performed. The update has been requested but may not have already been performed. Nevertheless, if the update has been performed, the quality of the answer is enhanced since the value is up to date.

**Characterized aggregate function**

Conventional relational DBMSs use some heuristics in the execution plan. As an example, selections are performed before joins since the natural complexity of the selection is linear (i.e., no index is used) and since the natural complexity of the join is quadratic. We propose to use a similar heuristic.

**Definition H3:** In a time-constraint environment, we propose to give priority to characterized aggregate function that are classified as linear, then as quadratic and then as super.

Obviously, this is only a heuristic since we may be in the same situation as the conventional relational DBMS. As an example, in some circumstances a join (formally defined as a selection on a Cartesian product) may be transformed into a (small) set of selections and therefore the complexity is now linear. So an aggregate function classified as quadratic may be transformed into a linear complexity depending on the amount of data during the evaluation of an execution plan.

**Lattice**

The lattice is defined to provide a better relevance between clients’ needs and the answer that could be provided. The lattice is a heuristic by itself since it provides a partial order within the different attributes involved in a query. This level is the highest level of heuristic. It provides an order between the different attributes that can be involved in the result. The lattice defines a partial order. A relative order within a level of the lattice is defined using H1, H2 and H3 otherwise, the order is not important. The order of application between H1, H2, and H3 is external schema dependent (i.e., view dependent).

**CONCLUSION**

Taking advantage of new applications provided by Location Based Services or GPS data requires introducing the notion of real time constraints in the DB models. Conventional DB models, extended to manage time and/or spatial data, provide an opportunity to manage real applications. A new step has to be defined to handle time constraint in the query resolution mechanism since Location Based Services often involve real-time constraint.

In this article, we proposed to enhance a DB model whatever the underlying formal model is (e.g., UML, object-oriented, object-relational). It takes into account the fact that the response time may vary depending on the context. The introduction of a partial order defines a relative importance between the set of data that may be provided as a result. The aggregate functions involved in a query may be or not computed depending on the available time. The evaluation is based on the computational complexity of the aggregate function. The coherency of a DB, mainly for navigation application is very important. Therefore, the specification of hard coherency maintenance is provided with the strong dependency. The Extended Functional Dependency enhances the relevance of the attributes provided in an answer.

Following the methodology proposed in (Fortunato et al. 2000), our new step will be the definition of the associated discrete model (i.e., to fix representations). Using these data definition language extensions, a query resolution model can be defined for a real time application context in order to present a sub-set of the data that should be given when no resolution time constraints are applied.
REFERENCES


BIOGRAPHY

MICHEL MAINGUENAUD was the head of the Engineering Information System department of the Institut National des Sciences Appliques de Rouen (INSA-Rouen) and then the Dean for Education of INSA-Rouen. He is currently the director of the CNRS/GDR2340 SIGMA-Cassini research group (www.sigma-cassini.org).
From a Master Server Architecture to a Feedback Control Architecture

Bechir Alaya, Claude Duvallet, Bruno Sadeg
LITTIS, UFR des Sciences et Techniques
25 rue Philippe Lebon, BP 540
F-76058 LE HAVRE CEDEX
email: Firstname.Lastname@univ-lehavre.fr

ABSTRACT

Today’s multimedia applications ask for a complex set of quality of service (QoS) requirements. These applications define time, bandwidth and synchronization constraints and manage large quantities of data. Due to the similarities existing between multimedia applications and real-time database systems (RTDBSs), we propose, in this paper, an approach which consists in exploiting some works related to the QoS management in RTDBSs in order to apply them to multimedia systems. We particularly propose a FCS-MS\(^1\) architecture, an improvement of the original master server architecture, which deals with QoS management by optimizing the resources use and reducing significantly the system overloads.

KEYWORDS
Distributed Multimedia Systems, Feedback Control Loop, Quality of Service, Real-Time Systems.

Introduction

These recent years, there are many researches which are not only interested to do minimize computation time, but also to satisfy application time constraints, i.e. deadlines, release times, etc. In this paper, we focus on a kind of these applications: multimedia applications. These applications must exchange very important quantities of data and they require the treatments to be done before fixed dates to guarantee an acceptable quality of service (QoS) in the streams presented to users. RTDBSs are the systems adapted to such data management while dealing with a certain QoS (Ramaanritham, 1993) (Ramaanritham et al., 2004).

Many distributed multimedia applications must face to predictable loads which cause the system overload. For example, user-demands may arrive burstly in a short period, which degrades the QoS provided to the user. Therefore, the need to design systems that provide a certain quality to users already served, has become a mandatory goal. In multimedia applications, the management of the QoS of the video packets allows to answer to these new needs. Since one decade, researchers try to adapt efficiently existing techniques to the video packets management without modifying the initial infrastructure. The main issue is the adaptation of the available resources (bandwidth, buffers size, video servers, etc.) and the proposition of new techniques to manage streams when instability periods (overload or under-utilization) occur. The goal is to assure an acceptable QoS provided by the system to users, while respecting the multiple requirements of the video streams.

Several studies have focused on the definition of mechanisms and strategies which allow the system to provide an acceptable QoS.

The main architecture proposed for QoS management in the distributed multimedia systems is the FCS-MS architecture which is based on a feedback loop (Wu et al., 2001). This approach was inspired from the work on QoS management in RTDBS, due to the existing similarities between the data management constraints in distributed multimedia systems and those of RTDBSs (Amirijoo et al., 2006) (Kang et al., 2002a).

First, we present the multimedia system architecture that we use. Then, we present the new master server architecture which we have developped to allow the QoS stabilization during the overload periods, more especially when the user-demands arrive in sudden manner. Finally, we present and comment the results simulation we have done to test the validity of our approach. We conclude this article by recalling the main ideas and by giving some perspectives to this work.

QoS in multimedia applications

QoS and distributed multimedia applications

QoS in a multimedia applications may be defined as the requirements in terms of bandwidth, quality of visualization, delay, and rate of video packets loss. Our approach consists in taking into account researches already done on the management of QoS in RTDBSs (Kang et al., 2002b) (Amirijoo et al., 2003) and their adaptation to multimedia systems. To this purpose, we propose an adaptation of a method based on feedback control architecture to distributed multimedia systems (Dulgheru, 2004).

\(^1\) Feedback Control Scheduling for Multimedia Systems.
This adapted method is called FCA-DMS (Feedback Control Architecture for Distributed Multimedia Systems). Our approach consists in controlling the multimedia system congestion by discarding or not some clients requests of certain types according to the system state, notably to the video server capacity. The simulations we have done showed that this approach improves the QoS provided to users.

Feedback control architecture for QoS management

In a previous work, N. Dulgheru has proposed an architecture, named QMPEGv2 (Dulgheru, 2004) (ISO/IEC 13818-2, 1994) (Ng et al., 2000) which deals with distributed multimedia systems (cf. figure 1). The architecture proposed is composed of three main parts:

- A master server: It accepts requests from clients, chooses the video servers able to serve the demand, supervises the system state and adjusts the video streams in order to maintain the QoS initially fixed.

- Video servers: they send the video packets to the clients under the master server control.

- Clients: they send requests to the master server and receive the video frames from the video server. When a state change occurs, they send a feedback report to the master server.

When a video on demand is requested, the following steps are executed according to FCA-DMS architecture:

1. A client sends a request to the master server to get a video with a certain level of QoS;
2. The master server broadcasts the request to the video servers available in the system;
3. The video servers send back their responses to the master server, which chooses one among them;
4. A stream is opened between the chosen video server and the concerned client;
5. The master server asks the video servers to adapt their QoS, when necessary.

The feedback loop is used when there is a need to adapt the QoS to the load system conditions: it observes the QoS obtained by the client and, if necessary, it asks the concerned video server to improve it.

QoS degradation and feedback control loop

In order to improve the QoS, the system increases or decreases the number of transmitted frames of certain types. To this purpose, we based our action on the characteristics of the standard MPEG format (ISO/IEC 13818-2, 1994), that defines a mechanism to code frames at the time of the video compression. When a video sequence enters the system, it is compressed and coded according to following three types of frames: Intra frames or I-frames (reference frames), Predicted frames or P-frames (which allow to rebuild a frame using an I frame) and Bidirectional frames or B-frames (which use I frames and P frames to rebuild a sequence). So, I frames are the most critical frames. In order to reduce the eventual resource overload, our approach consists in removing some frames from a video sequence in a controlled manner. We use a feedback loop which allows to stabilize the system during the instability periods (Bouazizi et al., 2005).

The approach is based on two principles: observation and auto-adaptation. The observation principle consists in observing the results obtained by the system and checking if the current QoS observed is consistent with the QoS initially required, e.g. in VoD application, the system checks if the video sequences are presented to users without interruptions. The auto-adaptation consists in adapting the results according to the QoS required by the clients, by adjusting some network and video parameters, e.g. the system increases or decreases the number of accepted frames. The feedback loop ensures then the system stability. We propose in the following paragraph a new architecture of the master server based on the feedback control in order to control the QoS provided to users in case of multimedia system overload.

How the master server works?

Master server architecture

In the master server architecture, the monitor allows to collect the remaining capacity of every Video Server (VS) and distinguishes between the following two types of customer demands:

- a new video.
- a QoS modification.

We are interested by the QoS modification requests, which are more critical. Indeed, new video demands will use more resources and will probably disrupt the video packets transmission, decreasing then the QoS provided to customers. According to the values of the remaining video servers capacities, the admission controller determines whether or not a demand will be accepted in the system. This control is done by comparing the capacity required by the task and the capacity provided by the video server. This value $E$ is given by:

$$E = \text{capacity(VSV)} - \text{capacity(demand)}$$
Figure 1: Functional model of the FCA-DMS architecture.

Figure 2: Adapted feedback loop for multimedia applications

Where:

*capacity (VSV): capacity of video server.*

*capacity (demand): capacity needed for a new demand.*

If \( E \geq 0 \) then the system accepts the demand else
If \( E < 0 \) the demand is aborted by the system or it is presented if it exist an other VS not saturated.

The QoS controller task takes the decision to increase or to decrease the QoS, according to the comparison of the current demand value with the last value measured.

- **Decreasing of QoS:** When the QoS controller decides to decrease the QoS, no problem occurs. It sends immediately a message to concerned VS and asks them to reduce the QoS. The advantages here are (1) to increase the server capacity on one hand, and (2) to free some resources that might be exploitable in case of QoS increase demand, in the other hand.

- **Increasing of QoS:** the video server in charge of a video may not find enough resources to answer to the customer demand. For example, if the VS ca-
Capacity is 20 frames by second (f/s) and three customers ask a QoS of 15 f/sec of the same VS, the probability of the load of the system increases. The admission controller has authorized this demand to enter the system because it has checked that it exists one or several not saturated VS.

**Replication strategy**

A video server can distribute only videos stored on its disks. If a video is not accessible on several servers [Wolf and al. 95] (one VS contain this video), the probability that this last VS will be saturated increases. Therefore, it is necessary to define a new distribution strategy (Algorithm 1) of video packets in order to have another video server which is used to answer to the customer request. The saturated video server sends a request to the nearest video servers. Each video server behaves according to one of the following three scenarios:

1. It possesses the video and it is able to treat the request (it is not saturated).

2. It possesses the video but it is unable to treat the request (it is saturated).

3. It doesn’t possess the video, but it is probably able to treat the request because it is not saturated.

In the two first cases, the replication strategy is not established. In the last case, the case manager, that has to control replication, sends an order to the saturated VS to start the replication. Consequently, the case manager elects a VS among those that answered and that are not saturated. The choice of the VS is done in order to get the best possible QoS. The demand returns again to the monitor, in order to allow to terminate the replication, then the monitor restarts works.

**Simulations**

**Presentation of the simulator**

To test the feasibility of our architecture, we had designed a simulator in order to verify the system behavior and its adaptation in relation to load variations. This simulator takes the architecture components presented in the Section. A master server is placed to video servers participating to the video diffusion. Thus, this service permits to organize the server addition while assigning them a number and referencing the accessible objects. After the master server start, the video are available in
The video distribution references at the master server and gets a number. To make a request, the client program have to ask to the master server, that distributes the request to available video servers. When the master server designated a video server able to answer to the client request - indeed that it arranges the asked content and it is able to provide the required QoS - it sends the video server reference to the client. Then, the streaming from the video server may begin. After some time of streaming, a return on the quality of service received by the client is sent to the master server. If it does not exist an available VS for the new video demand, the MS manages the QoS improved demand. It is based on the replication strategy in order to improve the

**Figure 4:** received frames rate without and with replication strategy.
QoS requested by the client. This simulator has been implemented in JAVA language and a object modeling language to design the three parts of the architecture.

**Simulation objectives and results**

Simulations have as main goal to show, how our simulator allows to efficiently adapt the QoS of client according to the current system load. Notably, the system must adjust when the client number that effect requests varies in the time. System overload comes from different sources; for example, a large number of clients making requests in our system. We can limit this client number by a monitor localized at the master server level. Also, incontrollable use of the network such as, network resource used by other applications can cause the congestion. Our architecture adjust itself while reducing the QoS stored to the different client.

In figures 4, 5 and 6, we have three curved lines G1, G2 and G3 that indicate:

- **G1**: received frames rate.
- **G2**: lost frames rate.
- **G3**: useful frames rate.
In theses figures, we noticed a difference in lost frame rate and received frames rate to the customer and useful frames rate. In the figure 4(a), we observed lost frames rate, received frames rate and useful frames rate before using the replication strategy of video contents in case of system overload.

In the figure 4(b), we noticed the same rate after the implementation of replication strategy. After sometimes, we notice in the figure 4(b) that loss frames rate is decreasing also increasing the received frames rate and useful frames rate. We have not noticed an important gain for the received frames rate and the useful frames rates level, because our replication strategy is not sufficient to stabilize desire QoS by the client. But, by combining our replication strategy with other techniques such as bandwidth sharing also the storage technical of the VS, we can have very important gains in received frames terms. Our simulation shows almost 10% of gain of the loss frames rate. In the new MS architecture, frames loss become somewhat controlled, since, our new architecture is also based on observation and auto-adaptation technique and on the principle of demand selection.

Conclusion and futures works

While current resource management systems provide mechanisms which provide reliability with respect to QoS, it is not sufficient since there are many well established application scenarios where QoS management is required, e.g., distributed multimedia systems. Our main contribution is related to the adaptation of master server architecture and the establishment of a video replication strategy. A possible extension of this work is the enhancement of the architecture, that we have presented, in order to bring some fault tolerance because of the presence of only one master server.

We have also presented the importance of the master server architecture improvement and have given a certain priority to the QoS modification demand, in order to increase its reliability and robustness and to converge towards the QoS specified by the client.

Simulations results allow us to validate the feasibility of our approach and should allow to provide results demonstrating the real contribution of this new approach.

An other possible future work would consists in building a real video on demand server based on the architecture that we propose. We will take into account of frames storage management and frames organization in video servers, notably, the most efficient manner (from QoS point of view) to videos broadcast between the different video servers. This work requires to compare the performances obtained when using each manner to organize and store videos.

REFERENCES


Ng J.; Leung K.; and Wong W., 2000. Quality of Service for MPEG Video in Human Perspective. Tech. rep., Hong Kong Baptist University.


Design and Analysis of Computer Experiments with SimExplorer

Florent Chuffart, Thierry Faure, Nicolas Dumoulin, Romain Reuillon, Guillaume Defuant
CEMAGREF – LISC
24 avenue des Landais - BP 50085
63172 Aubière cedex
email: firstname.lastname@cemagref.fr

KEYWORDS
Design of Experiment, Sensibility Analysis, Simulation, Distributed Computing, Framework, IDE, Tool Kit.

ABSTRACT
SimExplorer is a framework that includes an API\textsuperscript{1} and an IDE\textsuperscript{2} dedicated to design and analysis of computer experiments. SimExplorer combines blind computing on demand service access, traceability and reporting features and sensitivity analysis tools. This tutorial provides the concepts and exemplifies a typical use of SimExplorer, performing a global sensitivity analysis of a given individual-based predator prey model.

CONTEXT
The constant growth of computing power gives the possibility to run simulations of an increasing complexity. This is particularly the case for environmental dynamics modelling, which often involves coupled heterogeneous sub-models (biological model coupled with meteorological or hydrological models for instance) or individual-based models which represent explicitly individuals of a given population (human, plants, animals, \ldots). One can expect that this growth of the model complexity will provide a higher accuracy and a better understanding of the phenomena at stake. Unfortunately, this expectation is very often deceived, because the computer models become themselves so complex that it is more and more difficult to understand them, and to evaluate the confidence to give their results. Consequently, their use as decision supports is seriously weakened, because decision makers need understanding and confidence to ground their decisions. This situation suggests to consider these models exactly like complex natural phenomena, that scientists try to understand. The most important difference is that to design experiments on a computer model is generally easier than on a natural phenomenon [6]. Generally, it is necessary to develop a specific application to “explore” each model. This application should often manage the generation of well controlled initial conditions: specifically distributed lists of values, for example: spatial distributions, networks of interactions with specific properties (small world, scale free, \ldots). Such an application represents heavy investments, and much of the scientific quality relies upon it. The motivation behind SimExplorer [3, 4] is to invest once for all in a tool that can be used to design and execute properly experiments on any model. The goals are multiple. First, we want to externalise the development of the model exploration, in order to make available some generic methods and tools which can be applied in most of the cases for any model to explore; next, to increase the factorisation of available components, and therefore lower the investment for good quality model exploration applications; finally to facilitate a quality insurance approach to model exploration using traceability features.

TYPICAL COMPUTER EXPERIMENT
A typical computer experiment integrates sampling step, model evaluation and result analysis (figure 1, top). The sampling step computes each scenario that will be evaluated by the model. Next, results are analysed and experimenter could watch indicator evolution according to input parameters previously sampled. That scheme represents the most cases of computer experiments but design and analysis of computer experiment could be more complex. For example, the figure 1 shows a workflow where results are injected into the sampling step to explore a specific part of the model factor domain.

\footnotesize{API - Application Programming Interface. 
IDE - Integrated Development Environment.}

\begin{center}
\textbf{Figure 1: Basic and iterative workflows.}
\end{center}
on the figure 2. The left side of the window allows to visualize and edit the workflow of the exploration application. In the middle, editors are available according to the component selected in the workflow. The right side of the window presents several tools to define the data used in the exploration runtime.

**Sampling Tool** – An experimental design aims to explore a model upon a specific domain. The Sampling Tool is in charge of selecting elements of this domain according to the selected design.

**Model Launcher** – Model exploration requires expensive model execution in CPU and/or memory. The model launcher ensures the distribution of this task over cluster or grid architecture.

**Statistical Libraries** – Numerical exploration of model produces large amounts of data, that need to be aggregated or graphically displayed to be interpreted. Third party libraries could be plugged in the SimExplorer runtime to improve the available analysis tools. SimExplorer integrates the access to the R³ software and Dakota⁴, and other numerical tools integration is planned.

![Sampling Tool and Core Engine dialog.](design.xml)

**Reporting Features** – In order to disseminate and communicate about and experiment, SimExplorer offers reporting features. Reports are elaborated by collecting data and metadata from the information system, and organising them into a document format (web, pdf …).

**Information System** – The information system stores the steps and results of the exploration, and can be synchronized with a server application that allow to share data with other users. This component is relevant to ensure the reproducibility of any exploration and to improve the collaboration and the publication of results.

**Service Directory** – SimExplorer is based on Service Oriented Architecture. Consequently, the Service Directory component ensures service discovering through the network.

**Core Engine** – The core engine drives services according to the design. It uses Service Directory to discover service and access to their methods. For example, figure 4 illustrates the dialogue between de Sampling Tool and the Core Engine. The Core Engine address the Sampling Tool using design specifications. In response, the Sampling Tool gets asked scenarios that will be used to evaluate the model.

3R – [http://www.r-project.org/](http://www.r-project.org/)

---

**SIMEXPLORER FRAMEWORK**

SimExplorer is developed with a component based architecture in Java (OSGi[5]). The graphical user interface (GUI) is based on the NetBeans Rich Client Platform, which includes some pluggable tools to help the development of experimental designs, and language editors (Java, Groovy, Ruby). Figure 3 illustrates the basic components of SimExplorer architecture.

![SimExplorer architecture.](design.xml)

**IDE** – The integrated development environment (IDE) offers graphical user interface, and programming script language. Its purpose is to design an exploration application, that is achieved by building a workflow of treatment components. A screen shot of the IDE is showed on the figure 2. The left side of the window allows to visualize and edit the workflow of the exploration application. In the middle, editors are available according to the component selected in the workflow. The right side of the window presents several tools to define the data used in the exploration runtime.

**Sampling Tool** – An experimental design aims to explore a model upon a specific domain. The Sampling Tool is in charge of selecting elements of this domain according to the selected design.

**Model Launcher** – Model exploration requires expensive model execution in CPU and/or memory. The model launcher ensures the distribution of this task over cluster or grid architecture.

**Statistical Libraries** – Numerical exploration of model produces large amounts of data, that need to be aggregated or graphically displayed to be interpreted. Third party libraries could be plugged in the SimExplorer runtime to improve the available analysis tools. SimExplorer integrates the access to the R³ software and Dakota⁴, and other numerical tools integration is planned.

3R – [http://www.r-project.org/](http://www.r-project.org/)
SimExplorer platform is currently tested on the exploration of complex models from various applications fields (bacterial colony model, savannah and language dynamics) in the European project PATRES (NEST-043268) context.

OUTLINE OF THE TUTORIAL

The attendees will acquire the concepts and will be shown some practical examples. The tutorial begins with an overview the the SimExplorer software, and it introduces the main methods for designing experiments. Then, it describes in more details an example of model exploration based on the predator-prey model (described in the next section). This example will particularly illustrate the potential of SimExplorer IDE features. Then, it will show examples of the client-server information system features, the distributed execution functionality and to the report generation tool. Finally, if it is technically possible, the attendees will be assisted to try SimExplorer on their own model.

PREDATOR PREY MODEL WITH FIGHTS

The Predator Prey model with fights we consider (HDPP) is an individual-based version of a particular Lotka-Volterra predator prey model [1, 2], in which the predators can adopt aggressive or cooperative tactics to share the preys. It integrates individuals – predators and preys – that are spatially located on a 2D-grid. At each time step, all predators make several random moves, of a limited range. If they visit a site of a grid where a prey is located, they capture the prey. Then, they can share or struggle for the prey with other predators are located within a given neighbourhood. Their attitude (aggressive or cooperative) is defined for the time step. But after each time step, the predators can change their attitude by observing the success of the attitude of their predator neighbours during the last time step. They tend to adopt the most successful attitude.

The HDPP model analysis reveals that, depending on the values of its parameters, the model dynamics can lead to the extinction of the predators, or a single attractor, or two attractors, or one attractor and the extinction. Figure 5 exemplifies the attractor, trajectory, partition concepts for a given state space. State space is divided into a regular grid. Each cell is characterized by the trajectory starting from its centre. When the trajectory get out of the cell, the new cell where the trajectory is, is associated to the starting cell as its next. Thus, the state space is represented as graph where circuits are attractors, elements that compose two path to the same attractor are in the same partition.

Individual behaviour of the HDPP model is classically characterized by the mortality and birth rate of each species and the prey carrying capacity parameter. Due to the aggressive and cooperative tactic introduction, model becomes sensitive to the gain parameter. Finally, individual based model needs the neighbourhood size parameter.

Figure 5: HDPP model portrait of phase with cycles.

CONCLUSION

The SimExplorer software is freely available under GNU GPL licence on http://www.simexplorer.org.

REFERENCES


LATE PAPER
The GeoCells project

Patrice Langlois
UMR CNRS IDEES (6228), University of Rouen (France)
patrice.langlois@univ-rouen.fr

Dominique Goyat
UMR CNRS IDEES, University of Rouen (France)

Bernard Elissalde
UMR CNRS IDEES (6228). University of Rouen (France),
bernard.elissalde@univ-rouen.fr

Keywords: Mathematical simulations, spatial analysis, cellular automata

Introduction

The purpose of the GeoCells project is to create a modelling platform, supported by geographical information layers. Its main motor is a geographical model generator based on topological cellular agents. Starting from the description of a model considered as representative of a problematic in association with a real or virtual geographical environment, throughout this platform, the user builds an automated structure that enables to make this model live in order to see its evolution and to verify its level of pertinence.

The cellular automata GeoCells is based on interrelated processes between variables (like time periods, growth rates in the GDP per head, flows of public investments) and three geographical levels (European level, national level and regional level). This three levels are used to lay the emphasis on the fact that the European Union structural expenditure, and its spatial impact, work as a rules-based system in which three levels are interacting. When the cellular automata is launched, according to the rules we chose, the application work to set up vertical and horizontal links between levels, variables and spatial units. Simulations were made in order to evaluate on the one hand, the specific role of each level and each variable, and on the other hand how some change in one part affects the whole system.

1. From the strict notion of cell to the complex spatial agent’s one

The GeoCells project is part of the continuity of two experiences developed by our laboratory (MTG Rouen, UMR IDEES) [Langlois and al 02], [Germond and al 03], [Germond and al. 04].

First of all the SpaCelle software, which is a cellular automaton based on rules dealing only with qualitative variables, followed by another very different cellular model, allowing to simulate continuous flows such as surface flow in hydrology called the RuiCells model.

The strict formal concept of cellular automata defines a cell as purely reactive. A cell in the broad sense does not know how to do anything else but to change its internal state according to the perception of its environment and of its own state. Nevertheless, in a wider sense, the notion of cell can be understood in its analogy with the living world. Moreover, its based on this analogy that von Neumann defined the term of cellular automaton, in order to try to model the phenomenon of living agents’ reproduction. In this sense, a cell is not only reactive, but becomes an agent which can own a complex behaviour. A cell must be delimited in space by its membrane, which is both a frontier separating the inside from the outside, and a filter controlling solid, energy, and information flows with its environment.

In addition, we often need to define spatial units of any forms, because this is the way that geographical information is available: agricultural plots, land registry, urban units, etc. Thus we have tried to define directly a behaviour (a dynamic) to spatial units of any forms. The polygonal form must thus be directly usable as a geometric support of a cell in a geographic simulation system.

1.1. The GeoCells project, a multi-layer automaton

The purpose of the GeoCells project is to create a simulation platform based on geographic information layers. Its main driving force is a geographic model generator based on topologic cellular agents.

From a model description considered as representative of an issue in association with a real or virtual geographic environment, the user builds,
through this platform, an automated structure allowing this model to live, in order to see its evolution and to check its pertinence level.

The overall functioning principle of GeoCells is given by the following diagram:

![Diagram of GeoCells functioning principle]

Figure 1. GeoCells functioning principle

The different distinct parts visible on the diagram match the following stages:

1. The user calls for a set of static information used as a support to the model and thus describing its initial physical environment.
2. The user, relying on the above data, provides the desired model dynamic, that is to say the group of action features peculiar to the simulation and relating to the simulator-user interaction,
3. GeoCells translates this information and generates an automated model and a command interface,
4. The user communicates by means of the interface in order to change certain parameters relating either to the simulation or else to the ones linked to the results display.

2. Basic principles

a) Cellular structure

The system is based on a group of geographic information layers. Each layer is made of features from a same class. To each layer of information, and to each feature of this layer, matches a cell, which main asset is to own, in addition to the feature’s physical components (location, shape, size...), the knowledge of its neighbourhood and above all its behaviour dynamic.

Each cellular class owns:
- Behaviour rules giving to the cells of its class the same function in the system (district, plot, department),
- properties and attributes (perimeter, surface area, budget...),
- relations with cells from other layers of the system.

b) Hierarchized structure

The system takes into account the hierarchical relations existing between layers, such as the ones explained on the diagram below (a district [layer 1] belongs to a department [layer 2] and the department is made up of districts...), but nothing prohibits to implement other relations between cells of different layers, such as for example transport connections, etc.
c) Cells characteristics

Behaviour - It is different depending on the cellular layer to which they belong. Within their own layer, by virtue of their peculiar attributes, their impact on the environment can be very variable according to the model. Another specificity is that they can also exchange with other cells from all of the automaton’s layers. These exchanges are carried out in the form of flows.

States - Each state represents an attribute which is likely to evolve during simulation. Therefore, they can each take on a qualitative or quantitative nature. Moreover, each state, through exchanges, in their interactions with their neighbourhood, can have a different purpose, informational for qualitative but also informational or energetic for the two other natures.

We use the term « energetic » for flows or physical magnitudes likely, of course, to vary during simulation, but always by respecting the best a conservation principle, dear to physicians. In an economic-sounding simulation, we consider therefore a financial flow as an energetic flow, even if conservation is a complex notion in Economy. In contrast, magnitudes or informational flows are considered as no energy.

Life stages - Any cell owns a life pace that can be broken down into four stages:
- input reading (incoming exchanges),
- execution of its action program (its behaviour),
- output writing (outcoming exchanges),
- context storage. Each cell must at least keep track of the previous context’s content.

Capacity - These last cells have their own attributes and their own behaviour rules that can generate simple actions or conditioned actions, with more or less broad evaluation potential. In other words, they can own an “intelligent” behaviour tantamount to the agents ones. Nevertheless, we stick to the term of cell.

Let’s note that there are no theoretical limit as for the number of cells that can be managed by the system. Only the hardware’s possibilities may limit the simulation’s achievement.

2.1 Generated automata characteristics

Time definition - In order to scan all cells of all layers, therefore to give them the means to carry out their actions, the system takes a certain (physical) machine time that depends on the performances of the host computer. This proceeding time matches the lapse of time which is the system’s temporal reference (logical time). We can therefore build from this reference a time unit adapted to each simulation.

Synchronization – One of the difficulties of this type of mechanism is to maintain the temporal coherence between every cellular layers. In order to do so, the automaton synchronizes the system by ordering itself the starting of each cell’s life stage. This synchronization applies to:
- all of the cell’s input reading (incoming flows),
- the execution of all of the cell’s peculiar processes, with according to the circumstances, change of its internal states
- all of the cell’s output writing (outcoming flows).
- saving of all current contexts.
Communication canals - An unidirectional communication canal feature was introduced when it was necessary to implement the system’s multilayer nature with the flexibility required by exchanged data’s multiform nature, combined, according to the models, with the possible plurality of flows. This feature is automatically generated and sized. Each cell owns the input and output references relating to the canals that concerns it, and this is true for each communication action (informational and/or energetic). For this reason, it has the knowledge of its environment and enters into dialogue with it.

Command interface - According to the needs, each of the model’s influential magnitude is combined with an interface component, for example, in the form of a cursor, giving the opportunity at initialization to change its value. All of these components, created dynamically, associated to a certain number of structural dialog components (menus, etc.), makes the system’s command interface, the simulator’s control panel.

2;2GeoCells’ field of action

GeoCells can be used in any model that brings into play phenomena where spatial component is predominant. Among these phenomena, we can cite the diffusion or the propagation of magnitudes (physical or not), whether they are generated by rules of neighbourhood-contact (ex : frontiers between cells of a same layer) and/or by rules of neighbourhood-transport, by inserting between layers one ore more transport layers (linear topology) and access relations that are associated with them. We can also take on rules relating to virtual exchanges of a hierarchized type or not. In the following example, we carry out several hierarchized economic exchange flows.

The GeoCells model applies to the evolution of the GDP per capita in the Europe of the 15 and to the influence, in this evolution, of the game of the various aid granted as regional policy (ERDF). The platform is made up of three topological layers:

- Administrative regions level NUTS2 (512 cells : 511 regions + 1 cell representative of the outside)
- Countries (17 cells : 17 country cells + 1 cell rest of Europe + 1 « external » cell)
- EU (3 cells : 1 cell Europe of 15 + 1 cell rest of Europe + 1 « external » cell).

![Figure 3. Cellular layers](image)

The automaton takes into account the entire group of cells but makes interact only the wanted cells (in light colors) and was tested only over the European Union’s regions NUTS2 with 15 State members.
3. Description of the growth-diffusion model between European regions

We will now clarify the diffusion model a bit unusual that we have used. Let’s note \( X_i \) the GDP of the region \( i \), \( P_i \) its population and \( Y_i = X_i/P_i \) its GDP per capita at a moment \( t \).

We put forward the following hypothesis. Each cell has the aim to homogenize, through time, its standard of living \( Y \) in relation to its neighbours. But the standard of living is not comparable to a physical magnitude capable of diffusing like a flow. Its through the variation of wealth \( (X) \) symbolized by the GDP (by internal growth and by diffusion) or through the variation of population \( (P) \) (also by internal growth or by migrations) that each region can work in order to achieve its goal. In this model, at first approximation, we have considered that population was constant throughout time. It is therefore here only on the variation of \( X \) that relies the diffusion mechanism in order to reach the goal.

Another hypothesis is to consider that only a small fringe close to the borderline (area in dotted line on the figure) takes part in the diffusion of wealth, by the levelling-out of standards of living of the two neighbouring border fringes. Since we do not have any information on the spatial distribution of the populations inside a region, we must put forward the hypothesis of an uniform distribution. Consequently, instead of launching forth into geometric calculations of insane zoning, we use a simple proportionality parameter, accessible in the user interface by a cursor, called diffusion rate, which sets the rate \( k \) (of surface area, population, and wealth all at once, since we consider them as uniformly distributed over the region’s surface area), which takes part in the diffusion between regions. This rate defines therefore the part of the region’s surface area matching the border fringe, in which the standard of living is going to attempt to level up through time, with the neighbouring regions homologous fringes.

In order to model the diffusion between two regions \( i \) and \( j \), we then introduce the coefficient \( k_{ij} \) which is the surface area’s proportion \( i \) matching the intersection between the border fringe defined by \( k \) and the proportion \( p_{ij} \) of its land borderline

\[ p_{ij} = \frac{l_{ij}}{\sum_{k \in \operatorname{fais}(i)} l_{ik}} \]

, where \( l_{ij} \) is the borderline’s length between \( i \) and \( j \).

We then have: \( k_{ij} = k p_{ij} \)

If the wealth on the two sides of the border fringe between \( i \) and \( j \) was evenly distributed like connected vessels, we would obtain a levelled-out standard of living (which is not the average of the two previous standards), defined by:

\[ Y_{ij} = \frac{k_{ij} X_i + k_{ji} X_j}{k_{ij} P_i + k_{ji} P_j} \]

We can then define the variation \( dX_{ij} \) (positive if it emits or negative if it receives) of the diffusion from the region \( i \) towards the region \( j \) during a short lapse of time \( dt \) as being proportional to the concerned population \( (k_i P_i) \) and proportional to the difference between the current standard of living \( (Y_i) \) and the (local) aim of levelling-out \( (Y_{ij}) \) of standards of living \( i \) and \( j \). This can be translated into the following equation:

\[ \frac{dX_{ij}}{dt} = K k_{ij} P_i (Y_i - Y_{ij}) \]

The value of \( K \) is set internally (since we can already play on \( k \)).

By adding the border fringes of the region \( i \), we note down:

\[ dX_i = \sum_{j \in \operatorname{fais}(i)} dX_{ij} \]

One should notice that this diffusion is, by building, preservative of the mean \( \sum_{i=1}^{n} X_i \) (because one can verify easily that for any couple \((i,j)\) we have \( dX_{ij} + dX_{ji} = 0 \))

Moreover, the variable \( X_i \) is subjected to an a priori exponential internal growth, \( \frac{dX_i}{dt} = C_i X_i \)

Internal growth is adjustable, either individually region by region through the attribute table, either on the whole as being the same for all regions with the help of a cursor present in the user interface.

The final growth-diffusion equation is thus given by:

\[ X_i(t + dt) = X_i(t) + (C_i X_i(t) + K k_{ij} P_i (Y_i - Y_{ij}))dt \]
The lapse of time for the discretization of growth and diffusion processes are small compared to redistributing flows, because they correspond to continuous processes. We have selected the month as lapse of time, that also matches the time unit that we chose, so $dt = 1$. ($C_i$ is then the twelfth of the annual growth rate).

The equation with this lapse of time is then written:

$$X_i(t + dt) = (1 + C_i)X_i(t) + K_j k_j P_i(Y_i - Y_j)$$

The model introduced here attempts to give an account of the crossed recursion of the processes’ effects contributing to regional dynamics : the region’s peculiar growth, the redistribution mechanism linked to its membership to a wider territorial group (State, Union), and finally neighbourhood effects. Many economic models try to isolate the various sectoral variables in the final growth. Here, we suppose that the three components presented above create system effects introducing a large part of uncertainty in terms of growth.

We will not describe in detail the part of the model concerning aid and contributions, insofar as they show through clearly enough in the settings of the user interface.

References:


Gross C., Opérations topologiques et géométriques sur les multicartes combinatoires ; application à la cartographie thématique, thèse Université de Strasbourg, 1989.


AUTHOR LISTING
## AUTHOR LISTING

<table>
<thead>
<tr>
<th>Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ababueh J.</td>
<td>196</td>
</tr>
<tr>
<td>Abelha A.</td>
<td>291</td>
</tr>
<tr>
<td>Adamko N.</td>
<td>312</td>
</tr>
<tr>
<td>Ahvenlampi T.</td>
<td>45/161</td>
</tr>
<tr>
<td>Alaya B.</td>
<td>566</td>
</tr>
<tr>
<td>Albert V.</td>
<td>7</td>
</tr>
<tr>
<td>AL Hawari S.</td>
<td>196</td>
</tr>
<tr>
<td>Al-Hudhud G.</td>
<td>267</td>
</tr>
<tr>
<td>Almeida P.</td>
<td>121</td>
</tr>
<tr>
<td>Al-Obaidy M.</td>
<td>201</td>
</tr>
<tr>
<td>Amador J.</td>
<td>385</td>
</tr>
<tr>
<td>Amantlon L.</td>
<td>556</td>
</tr>
<tr>
<td>Arfi M.</td>
<td>307</td>
</tr>
<tr>
<td>Ariotti P.</td>
<td>221</td>
</tr>
<tr>
<td>Avram D.</td>
<td>82</td>
</tr>
<tr>
<td>Ayeb B.</td>
<td>556</td>
</tr>
<tr>
<td>Ayesh A.</td>
<td>201/279</td>
</tr>
<tr>
<td>Baccouche L.</td>
<td>551</td>
</tr>
<tr>
<td>Balsamo S.</td>
<td>447</td>
</tr>
<tr>
<td>Belkadi K.</td>
<td>399</td>
</tr>
<tr>
<td>Bellemans T.</td>
<td>226</td>
</tr>
<tr>
<td>Bergeret M.</td>
<td>541</td>
</tr>
<tr>
<td>Bernot G.</td>
<td>423</td>
</tr>
<tr>
<td>Bertelle C.</td>
<td>258/264/524</td>
</tr>
<tr>
<td>Bisgambiglia P.</td>
<td>338</td>
</tr>
<tr>
<td>Blewitt W.F.</td>
<td>279</td>
</tr>
<tr>
<td>Bogdan L.</td>
<td>438</td>
</tr>
<tr>
<td>Brouin E.</td>
<td>338</td>
</tr>
<tr>
<td>Bruniaux P.</td>
<td>82</td>
</tr>
<tr>
<td>Burda R.</td>
<td>131</td>
</tr>
<tr>
<td>Camacho A.M.</td>
<td>51</td>
</tr>
<tr>
<td>Caris A.</td>
<td>391</td>
</tr>
<tr>
<td>Carneiro D.</td>
<td>175</td>
</tr>
<tr>
<td>Ceska M.</td>
<td>19/489</td>
</tr>
<tr>
<td>Chabrol M.</td>
<td>230</td>
</tr>
<tr>
<td>Charabidze D.</td>
<td>417</td>
</tr>
<tr>
<td>Chuffart F.</td>
<td>575</td>
</tr>
<tr>
<td>Cicirelli F.</td>
<td>479</td>
</tr>
<tr>
<td>Coen-Porisini A.</td>
<td>360</td>
</tr>
<tr>
<td>Colombo P.</td>
<td>360</td>
</tr>
<tr>
<td>Comet J.-P.</td>
<td>423</td>
</tr>
<tr>
<td>Cortés J.A.</td>
<td>150</td>
</tr>
<tr>
<td>Costa R.</td>
<td>175</td>
</tr>
<tr>
<td>Courdier R.</td>
<td>25/343</td>
</tr>
<tr>
<td>Courvoisier M.</td>
<td>7</td>
</tr>
<tr>
<td>Crane M.</td>
<td>348</td>
</tr>
<tr>
<td>Cutajar V.</td>
<td>19</td>
</tr>
<tr>
<td>Cutolo A.</td>
<td>238</td>
</tr>
<tr>
<td>Cybis H.B.B.</td>
<td>221</td>
</tr>
<tr>
<td>Daudé E.</td>
<td>535</td>
</tr>
<tr>
<td>David D.</td>
<td>25</td>
</tr>
<tr>
<td>Dealure Y.</td>
<td>348</td>
</tr>
<tr>
<td>Deffuant G.</td>
<td>575</td>
</tr>
<tr>
<td>Dessagne G.</td>
<td>107</td>
</tr>
<tr>
<td>Diaz M.</td>
<td>150</td>
</tr>
<tr>
<td>Dingle N.J.</td>
<td>137/145</td>
</tr>
<tr>
<td>Dormido S.</td>
<td>168</td>
</tr>
<tr>
<td>Drchal J.</td>
<td>191</td>
</tr>
<tr>
<td>Duchamp G.H.E.</td>
<td>264</td>
</tr>
<tr>
<td>Duckwitz S.</td>
<td>56/66</td>
</tr>
<tr>
<td>Dumoulin N.</td>
<td>575</td>
</tr>
<tr>
<td>Dupleac D.</td>
<td>433</td>
</tr>
<tr>
<td>Dupont D.</td>
<td>183</td>
</tr>
<tr>
<td>Dutot A.</td>
<td>524</td>
</tr>
<tr>
<td>Duvallet C.</td>
<td>566</td>
</tr>
<tr>
<td>El Medhi S.O.</td>
<td>456</td>
</tr>
<tr>
<td>Elissalde B.</td>
<td>581</td>
</tr>
<tr>
<td>Fachada N.</td>
<td>252</td>
</tr>
<tr>
<td>Faure T.</td>
<td>575</td>
</tr>
<tr>
<td>Feillée D.</td>
<td>107</td>
</tr>
<tr>
<td>Franzolini J.</td>
<td>516</td>
</tr>
<tr>
<td>Frattarulo R.</td>
<td>238</td>
</tr>
<tr>
<td>Furfaro A.</td>
<td>479</td>
</tr>
<tr>
<td>Galpin V.</td>
<td>492</td>
</tr>
<tr>
<td>Garrido D.</td>
<td>150</td>
</tr>
<tr>
<td>Georgiev V.</td>
<td>332</td>
</tr>
<tr>
<td>Gerzson M.</td>
<td>74</td>
</tr>
<tr>
<td>Ghnemat R.</td>
<td>258/264</td>
</tr>
<tr>
<td>Gomez-Elvira J.</td>
<td>28</td>
</tr>
<tr>
<td>Gourgain M.</td>
<td>230</td>
</tr>
<tr>
<td>Goyat D.</td>
<td>581</td>
</tr>
<tr>
<td>Guardères E.</td>
<td>541</td>
</tr>
<tr>
<td>Guinand F.</td>
<td>516</td>
</tr>
<tr>
<td>Hadi W.M.</td>
<td>196</td>
</tr>
<tr>
<td>Hill D.</td>
<td>107</td>
</tr>
<tr>
<td>Hoareau D.</td>
<td>343</td>
</tr>
<tr>
<td>Hofman R.</td>
<td>544</td>
</tr>
<tr>
<td>Huget M.-P.</td>
<td>343</td>
</tr>
<tr>
<td>Iacono M.</td>
<td>500</td>
</tr>
<tr>
<td>Janousek V.</td>
<td>467</td>
</tr>
<tr>
<td>Janssens D.</td>
<td>226</td>
</tr>
<tr>
<td>Janssens G.K.</td>
<td>373/391</td>
</tr>
<tr>
<td>Author</td>
<td>Page Numbers</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Jolly D</td>
<td>417</td>
</tr>
<tr>
<td>Kabuš W.</td>
<td>56</td>
</tr>
<tr>
<td>Kapron H.</td>
<td>412</td>
</tr>
<tr>
<td>Karahancer S.</td>
<td>56</td>
</tr>
<tr>
<td>Kausch B.</td>
<td>56</td>
</tr>
<tr>
<td>Khalil Z.</td>
<td>423</td>
</tr>
<tr>
<td>Kirchel L.</td>
<td>332</td>
</tr>
<tr>
<td>Klima V.</td>
<td>312</td>
</tr>
<tr>
<td>Knottedbelt W.J.</td>
<td>137/145</td>
</tr>
<tr>
<td>Köhler H.</td>
<td>35</td>
</tr>
<tr>
<td>Kohlmeyer J.</td>
<td>94</td>
</tr>
<tr>
<td>Kolář D.</td>
<td>473</td>
</tr>
<tr>
<td>Kovanik O.</td>
<td>247</td>
</tr>
<tr>
<td>Kubiak P.</td>
<td>183</td>
</tr>
<tr>
<td>Kumar B.</td>
<td>348</td>
</tr>
<tr>
<td>Květoňová S.</td>
<td>467/473</td>
</tr>
<tr>
<td>Lamotte W.</td>
<td>373</td>
</tr>
<tr>
<td>Langlois P.</td>
<td>581</td>
</tr>
<tr>
<td>Lebrecht A.S.</td>
<td>145</td>
</tr>
<tr>
<td>Leclaire P.</td>
<td>230</td>
</tr>
<tr>
<td>Leclercq E.</td>
<td>456</td>
</tr>
<tr>
<td>Lefebvre D.</td>
<td>456</td>
</tr>
<tr>
<td>Lefer W.</td>
<td>541</td>
</tr>
<tr>
<td>Lehmann A.</td>
<td>14</td>
</tr>
<tr>
<td>Leitold A.</td>
<td>74</td>
</tr>
<tr>
<td>Lemine B.O.M.</td>
<td>307</td>
</tr>
<tr>
<td>Lencse G.</td>
<td>38/404</td>
</tr>
<tr>
<td>Lewandowski A.</td>
<td>131</td>
</tr>
<tr>
<td>Licht T.</td>
<td>66</td>
</tr>
<tr>
<td>Lichtenegger G.</td>
<td>511</td>
</tr>
<tr>
<td>Limam S.</td>
<td>551</td>
</tr>
<tr>
<td>Lopes V.V.</td>
<td>252</td>
</tr>
<tr>
<td>Lopez-Herrero M.J.</td>
<td>369</td>
</tr>
<tr>
<td>Lucchini F.</td>
<td>258</td>
</tr>
<tr>
<td>Luigi Ratih L.</td>
<td>238</td>
</tr>
<tr>
<td>Machado J.</td>
<td>175/291</td>
</tr>
<tr>
<td>Mainguenaud M.</td>
<td>561</td>
</tr>
<tr>
<td>Mallet P.</td>
<td>524</td>
</tr>
<tr>
<td>Marek R.</td>
<td>327</td>
</tr>
<tr>
<td>Marin A.</td>
<td>447</td>
</tr>
<tr>
<td>Marin M.M.</td>
<td>51</td>
</tr>
<tr>
<td>Maroto J.</td>
<td>28</td>
</tr>
<tr>
<td>Marrone S.</td>
<td>500</td>
</tr>
<tr>
<td>Martínez F.</td>
<td>168</td>
</tr>
<tr>
<td>Martin-Villalba C.</td>
<td>168</td>
</tr>
<tr>
<td>Morvan G.</td>
<td>183/417</td>
</tr>
<tr>
<td>Muka L.</td>
<td>38/404</td>
</tr>
<tr>
<td>Nabaa M.</td>
<td>524</td>
</tr>
<tr>
<td>Neagu I.</td>
<td>82</td>
</tr>
<tr>
<td>Neves J.</td>
<td>121/175/291</td>
</tr>
<tr>
<td>Neves J.</td>
<td>175/291</td>
</tr>
<tr>
<td>Nguyen B.</td>
<td>317</td>
</tr>
<tr>
<td>Nigro L.</td>
<td>479</td>
</tr>
<tr>
<td>Nketsa A.</td>
<td>7</td>
</tr>
<tr>
<td>Novais P.</td>
<td>121/175/291</td>
</tr>
<tr>
<td>Novosad P.</td>
<td>489</td>
</tr>
<tr>
<td>Nyambo B.M.</td>
<td>373</td>
</tr>
<tr>
<td>Olivier D.</td>
<td>516/524</td>
</tr>
<tr>
<td>Paludetto M.</td>
<td>7/115</td>
</tr>
<tr>
<td>Pavel N.I.</td>
<td>433</td>
</tr>
<tr>
<td>Pecha P.</td>
<td>544</td>
</tr>
<tr>
<td>Pironti M.</td>
<td>299</td>
</tr>
<tr>
<td>Pretto C.O.</td>
<td>221</td>
</tr>
<tr>
<td>Prisecar I.</td>
<td>433</td>
</tr>
<tr>
<td>Provotolo D.</td>
<td>535</td>
</tr>
<tr>
<td>Pruzinec M.</td>
<td>35</td>
</tr>
<tr>
<td>Qaddoum K.S.</td>
<td>89</td>
</tr>
<tr>
<td>Ramaekers K.</td>
<td>226</td>
</tr>
<tr>
<td>Raud Z.</td>
<td>355</td>
</tr>
<tr>
<td>Remondino M.</td>
<td>299</td>
</tr>
<tr>
<td>Rempel M.</td>
<td>317</td>
</tr>
<tr>
<td>Reuillon R.</td>
<td>575</td>
</tr>
<tr>
<td>Richmond P.</td>
<td>213</td>
</tr>
<tr>
<td>Rodriguez M.</td>
<td>150</td>
</tr>
<tr>
<td>Romano D.M.</td>
<td>213</td>
</tr>
<tr>
<td>Romero G.</td>
<td>28</td>
</tr>
<tr>
<td>Rosa A.</td>
<td>252</td>
</tr>
<tr>
<td>Rubio E.</td>
<td>51</td>
</tr>
<tr>
<td>Sadeg B.</td>
<td>556</td>
</tr>
<tr>
<td>Sadeg B.</td>
<td>566</td>
</tr>
<tr>
<td>Sahraoui A.</td>
<td>115</td>
</tr>
<tr>
<td>Said A.H.</td>
<td>556</td>
</tr>
<tr>
<td>Santucci J.-F.</td>
<td>338</td>
</tr>
<tr>
<td>Schlick C.M.</td>
<td>56/66</td>
</tr>
<tr>
<td>Schmitz P.</td>
<td>66</td>
</tr>
<tr>
<td>Sebastián M.A.</td>
<td>51</td>
</tr>
<tr>
<td>Sebastien N.</td>
<td>343</td>
</tr>
<tr>
<td>Selberherr S.</td>
<td>380</td>
</tr>
<tr>
<td>Selmi C.</td>
<td>307</td>
</tr>
<tr>
<td>Author</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Sklenar J.</td>
<td>19</td>
</tr>
<tr>
<td>Skrbek M.</td>
<td>247/327</td>
</tr>
<tr>
<td>Snorek M.</td>
<td>191</td>
</tr>
<tr>
<td>Soler E.</td>
<td>150</td>
</tr>
<tr>
<td>Sroczan E.M.</td>
<td>521</td>
</tr>
<tr>
<td>Steffens T.</td>
<td>284</td>
</tr>
<tr>
<td>Studzinski J.</td>
<td>438</td>
</tr>
<tr>
<td>Subik S.</td>
<td>131</td>
</tr>
<tr>
<td>Sverdlov V.</td>
<td>380</td>
</tr>
<tr>
<td>Tackenberg S.</td>
<td>56</td>
</tr>
<tr>
<td>Tanguy .A</td>
<td>399</td>
</tr>
<tr>
<td>Thabtah F.</td>
<td>196</td>
</tr>
<tr>
<td>Torres J.</td>
<td>28</td>
</tr>
<tr>
<td>Urbaniak A.</td>
<td>112</td>
</tr>
<tr>
<td>Urquia A.</td>
<td>168</td>
</tr>
<tr>
<td>Uusitalo M.</td>
<td>45/161</td>
</tr>
<tr>
<td>Valentin J.</td>
<td>541</td>
</tr>
<tr>
<td>Verries J.</td>
<td>115</td>
</tr>
<tr>
<td>Vodozzo V.</td>
<td>355</td>
</tr>
<tr>
<td>Voessner S.</td>
<td>511</td>
</tr>
<tr>
<td>Wan F.</td>
<td>145</td>
</tr>
<tr>
<td>Wang L.</td>
<td>137</td>
</tr>
<tr>
<td>Wang Z.</td>
<td>14</td>
</tr>
<tr>
<td>Wets G.</td>
<td>226</td>
</tr>
<tr>
<td>Wietfeld C.</td>
<td>131</td>
</tr>
<tr>
<td>Wolters B.</td>
<td>284</td>
</tr>
<tr>
<td>Wörner A.</td>
<td>35</td>
</tr>
<tr>
<td>Wydra M.</td>
<td>412</td>
</tr>
</tbody>
</table>