10TH MIDDLE EASTERN SIMULATION MULTICONFERENCE

MESM'2009

EDITED BY

Professor
Marwan Al-Akaidi

SEPTEMBER 27-29, 2009

Beirut, Lebanon

A Publication of

EUROSIS-ETI

Printed in Ghent, Belgium

10TH MIDDLE EASTERN SIMULATION MULTICONFERENCE

SEPTEMBER 27-29, 2009

Organised by

ETI

Sponsored by

EUROSIS

De Montfort University

IEEE UKRI - SPC

Ghent University

Hosted by

Lebanese American University

Beirut, Lebanon

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

General Conference and Programme Chair

Professor Dr M. Al-Akaidi
De Montfort University,
Leicester, United Kingdom

Local Chair

Haidar Harmanani, LAU, Department of Computer Science, Byblos, Lebanon

Local Co-Chair

Nashaat Masnour, LAU, Department of Computer Science, Beirut, Lebanon

Local Committee

Faysal Abu Khuzam, LAU, Department of Computer Science, Beirut, Lebanon
Nashaat Masnour, LAU, Department of Computer Science, Beirut, Lebanon
Samer Saab, LAU, Department of Computer Engineering, Lebanon
Iyad Ouaiss, LAU, Department of Computer Engineering, Lebanon
Walid Keirouz, American University at Beirut, Department of Computer Science
Fouad Chedid, Notre Dame University, Lebanon

INTERNATIONAL PROGRAMME COMMITTEE

Modelling Methodology and Modelling Tools
John Dalseng, Finnmark College, Alta, Norway
Helge Hagenauer, Universitaet Salzburg, Salzburg, Austria
Brian Hollocks, Bournemouth University, Bournemouth, United Kingdom
Bjoern Johansson, Chalmers University of Technology, Goeteborg, Sweden
Panajotis T. Katsaros, Aristotle University Thessaloniki, Thessaloniki, Greece
Yan Luo, NIST, Gaithersburg, USA
Thomas O Mathuna, Cork Institute of Technology, Cork, Ireland
Michel Minoux, Universite P. et M. Curie-CNRS, Paris cedex 05, France
Tudor Niculiu, University Politecnica Bucharest, Bucharest, Romania
Marc Parenthoen, CERV, Plouzane, France
Zdravko Terze, University of Zagreb, Zagreb, Croatia
Dmitry Zaitsev, Odessa National Telecommunications Academy, Odessa, Ukraine

Analysis Methodology
Abdel Aitouche, Hautes Etudes Industrielles, Lille Cedex, France
Bruno Ciciani, Universita di Roma "La Sapienza", Rome, Italy
John Dalton, University of Newcastle, Newcastle-upon-Tyne, United Kingdom
Julie Dugdale, Universite Paul Sabatier, Narbonne, France
Mauro Iacono, SUN - II Universita di Napoli, Aversa (CE) Italy
Jens Kohlmeyer, Universitaet Ulm, Ulm, Germany
Francesco Moscato, Seconda Universita di Napoli, Aversa, Italy
Jairo R. Montoya-Torres, Ecole Nationale Superieure des Mines de Saint-Etienne, Gardanne, France
Nadia Zanzouri, ENIT, Tunis, Tunisia

Simulation of Networks and Communication
Salvatore D'Antonio, Consorzio Interuniversitario Nazionale per l'Informatica, Naples, Italy
Mahmoud Al-Qutayri, Khalifa University of Science, Sharjah, UAE
Med Salim Bouhlel, Sfax University, Sfax, Tunisia
Dirk Brade, FOI, Stockholm, Sweden
Eugen Dedu, LIFC, Montbelliard, France
Carlos Palau, UPV, Valencia, Spain
Djamel Sadok, The Federal University of Pernambuco - Brazil
Ahmad A. Sharieh, University of Jordan, Amman, Jordan
Veronique Veque, Universite de Paris Sud LRI, Orsay, France

INTERNATIONAL PROGRAMME COMMITTEE

Complex Systems Simulation
Marta Pla Castells, Universidad de Valencia, Valencia, Spain
Michele Colajanni, Universita di Modena, Modena, Italy
Krzysztof Fujarewicz, Silesian University of Technology, Gliwice, Poland
Tom Dhaene, University of Antwerp, Antwerp, Belgium
Jean-Christophe Guyot, EADS SPACE Transportation, Les Mureaux Cedex, France
Daniel Huber, Heinz Nixdorf Institute, University of Paderborn, Paderborn, Germany
Abder Koukam, Universite de Technologie de Belfort-Montbeliard, France
Jan Lemeire, VUB, Brussels, Belgium
Jorge M. Perez, ITP, S.Fernando de Hernares, Spain
Panos Periorellis, University of Newcastle Upon Tyne, Newcastle, United Kingdom
Fatima Rateb, University College London, London, United Kingdom
Marco Remondino, University of Turin, Turin, Italy
Marc Rouff, University of Caen Basse-Normandie, Caen, France
Jan Studzinski, Systems Research Institute of Polish Academy of Science

Simulation of Signal and Image Processing
Bruno Ciciani, Universita di Roma "La Sapienza", Rome, Italy
Aboul Ella Hassanine, Kuwait University, Safat, Kuwait
Tania Jimenez, LIA, Universite d'Avignon, France
Viorel Nicolau, "Dunarea de Jos" University of Galati, Romania
Bogdan Raducanu, UAB, Cerdanyola, Spain
Renate Sitte, Griffith University, Gold Coast Mail Centre, Australia
Joao Manuel R. S. Tavares, University of Porto, Porto, Portugal
Hanene Trichili, University of Sfax, Sfax, Tunisia
Ahmed M. Zeki, International Islamic University Malaysia (IIUM), Kuala Lumpur, Malaysia

Energy Systems Simulation
Paulo Maciel, UFPE, Recife, Brazil
Musleh B. AL-Otaibi, Alahmadi, Kuwait
Ahmad Jammal, Ministry of Education & Higher Education, Beirut, Lebanon
Janos Sebestyen Janosy, KFKI, Budapest, Hungary
Carlos Ramos, Polytechnic of Porto (ISEP/IPP), Porto, Portugal
Tzanko Tzanov, Polytechnical University of Catalunya, Barcelona, Spain
Patrick Waeger, EMPA, St. Gallen, Switzerland

Simulation in the Chemical and Petrochemical Industry
Ammar Al Khani, Espoo, Finland
Diganta Bhusan Das, Loughborough University, Loughborough, United Kingdom
Eugenio Ferreira, Universidade do Minho, Braga Portugal
Vahid Nassehi, Loughborough University, Loughborough, United Kingdom
Mohammad Al-Nawafleh, Tafila Applied University College, Al-Balqa' Applied University, Tafila, Jordan
Cornel Resteanu, Research Institute of Informatics, Bucharest, Romania
M R Riazi, Kuwait University, Safat, Kuwait

Multimedia and Virtual Reality Systems
Christos Bouras, University of Patras, Rion, Patras, Greece
Eckhard C. Bollow, Software Technology and IT-Project Management, Lueneburg, Germany
Maria Paule Ruiz del Puerto, University of Oviedo, Oviedo, Spain
Zhisheng Huang, Vrije University Amsterdam, The Netherlands
Christoph Laroque, University of Paderborn, Paderborn, Germany
Ahmed Mostefaoui , Laboratoire d'Informatique de Franche-Comte Montbeliard Cedex, France
Elena Pelican, Ovidius University of Constanta, Romania
Dorin-Mircea Popovici, Ovidius University of Constanta, Romania
Christian Reimann, C-LAB, Paderborn, Germany
Marco Roccetti, University of Bologna, Bologna, Italy
Zasriati Azla Sabot, University Tun Abdul Razak (UNITAR) Selanngor, Malaysia
Mike Sips, University of Konstanz, Konstanz, Germany

Decision Processing in Management
Hadhoum Boukachour, Le Havre University, Le Havre, France
Naoufel Cheikhrouhou, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
Joel Colloc, Le Havre University, Le Havre, France
Jeanne Schreurs, Universiteit Hasselt, Diepenbeek, Belgium
Christine Sybord, Lyon 2 Lumiere University, Lyon, France
Behrouz Zarei, Sharif University of Technology, Tehran, Iran

Modelling and Simulation in Biomedical Systems
Syed Afaq Husain, International Islamic University, Islamabad, Pakistan
Manuel Alfonseca, Escuela Politecnica Superior, Universidad Autonoma de Madrid, Madrid, Spain
Clemens Heitzinger, Arizona State University (ASU), USA
Manuel Gonzalez Hidalgo, Edf. Anselm Turmeda, Palma de Mallorca, Baleares, Spain
Tomislav Pribanic, University of Zagreb, Zagreb, Croatia
Oliver Straub, University of Ulm, Ulm, Germany
Jean-Pierre Thomesse, Institut National Polytechnique de Lorraine, Vandoeuvre Les Nancy, France
Olaf Tietje, Systaim GmbH, Zurich, Switzerland

Modelling and Simulation for Industrial Applications
Track Chair: Prof. David Stockton, de Montfort University, Leicester, United Kingdom
Luis Felipe Acebes Arconada, Universidad de Valladolid, Valladolid, Spain
Leopoldo Arnesto, UPV, Valencia, Spain
Michel Benne, Sucreries de Bois Rouge, Saint-Andre, France
Pascal Berruet, Universite Bretagne Sud, Lorient cedex, France

INTERNATIONAL PROGRAMME COMMITTEE

Christelle Bloch, Universite de Technologie de Belfort-Montbeliard, Belfort Cedex, France
PJ Byrne, DCU Business School, Dublin City University, Dublin, Ireland
A. Chatzinikolaou, Bosch Rexroth SA, Athens, Greece
Olaf Diegel, Massey University, Albany Campus, North Shore Mail Centre, Auckland, New Zealand
S Collart Dutilleul, LAGIS-EC-LILLE, Villenneuve d'Ascq, France
Georges Habchi, Polytech'Savoie, Annecy-le-Vieux, France
Sergej Kalaschnikow, VA TECH ELIN EBG Elektronik GmbH & Co, Vienna, Austria
Markus Koch, C-LAB, Paderborn, Germany
Habtom Mebrahtu, Anglia Polytechnica University, Chelmsford, United Kingdom
Pascal Meyer, Forschungszentrum Karlsruhe, Karlsruhe, Germany
Lars Moench, TU Ilmenau, Ilmenau, Germany
Carlos E. Pereira, Universidad Federal do Rio Grande Do Sul, Porto Alegre, Brazil
Caterina Rizzi, Universita degli Studi di Bergama, Dalmine, Italy
Amir Shirkhodaie, Tennessee State University, Nashville, USA
Hans Veeke, TU Delft, Delft, The Netherlands
Ramiro Velazquez, Universite Paris 6 - CNRS, Fontenay aux Roses, France
Val Vitanov, Cranfield University, Cranfield, United Kingdom
Ursula Wellen, Deutsche Post AG, Bonn, Germany
Joao de Silva Wellington, UFPE, Recife, Brazil

Simulation in Logistics, Traffic, Transport and Harbour Simulation
Paul Davidsson, Blekinge Institute of Technology, Ronneby, Sweden
Isabel Garcia Gutierrez, Universidad Carlos III de Madrid, Leganes, Madrid, Spain
Thomas Hanne, Fraunhofer-Institut ITWM, Kaiserslautern, Germany
Daniel Krajzewicz, GermanAerospace Centre, Berlin, Germany
Seung Man Lee, NASA Ames Research Center, Moffett Field, USA
Herve Manier, Universite de Technologie de Belfort-Montbeliard (UTBM), Belfort Cedex, France
Vittorio Maniezzo, Universita di Bologna, Bologna, Italy
Rafael J. Martinez, Universitat de Valencia, Valencia, Spain
Philippe Mussi, INRIA Sophia Antipolis, France Frederic Serin, IUT du Havre, Le Havre, France

Web based Simulation
Victor Bassilious, University of Abertay Dundee, United Kingdom
Jorge Marx Gomez, Otto-von-Guericke-Universitaet Magdeburg, Magdeburg, Germany
Wolfgang Kuehn, University of Wuppertal, Wuppertal, Germany
Jose Machado, Universidade do Minho, Braga Codex, Portugal
Mara Nikolaidou, University of Athens, Ilissia, Athens, Greece
Francesco Quaglia, Universita' di Roma "La Sapienza", Roma, Italy
Stefan Sarstedt, University of Ulm, Ulm, Germany
Alfonso Urquia, UNED, Madrid, Spain

Simulation in Information Processing
Julien Bourgeois, LIFC, Montbelliard, France
Christophe Claramunt, Naval Academy Research Institute, Brest Naval, France
Patrick De Causmaecker, University of Nottingham, Nottingham, United Kingdom
Khalil Drira, LAAS-CNRS, Toulouse Cedex 04, France
Laurent Guillon, Naval Academy Research Institute, Brest Naval, France
Jean-Claude Hochon, GFI Consulting, Toulouse, France
Majed Khodr, Manar University, Lebanon
Peter Lawrence, Monash University, Clayton, Australia
Christophe Roche, Universite de Savoie, Le Bourget du Lac cedex, France
Christian Siemers, TU Clausthal, Clausthal-Zellerfeld, Germany
Jim Torresen, University of Oslo, Oslo, Norway
Nagi Wakimm, Saint Esprit University, Beirut, Lebanon

Simulation in Fuzzy Systems, Neural Networks and Genetic Algorithms
Chrissanthi Angeli, Technological Education Institute of Piraeus, Athens, Greece
Piotr Bania, AGH - University of Science and Technology, Krakow, Poland
Mokhtar Beldjehem, St.Anne's University, Halifax, Canada
Ulrich Bodenhofer, Software Competence Center Hagenberg GmbH, Hagenberg, Austria
Paulo Alexandre Ribeiro Cortez, University of Minho, Guimaraes, Portugal
Rashidy Kanan, University of Amirkabir, Tehran, Iran
Nashat Mansour, Lebanese American University, Beirut, Lebanon
Antoni Ligeza, AGH - University of Science and Technology, Krakow, Poland
Marie-Ange Manier, Universite de Technologie de Belfort-Montbeliard (UTBM), Belfort Cedex, France
Paulo Moura Oliveira, UTAD-Universidade de Tres-os-Montes e Alto Douro, Vila Real, Portugal
Kifah Tout, The Lebanese University, Beirut, Lebanon
Angus Wu, City University of Hong Kong, Hong Kong

Simulation in the Built Environment
Bernadette O'Regan, University of Limerick, Limerick, Ireland
Agustin Yague, Universidad Politecnica de Madrid, Madrid, Spain

Simulation in Archaeology
Jean Yves Blaise, UMR CNRS/MCC, Marseille, France
Anders Hast, University of Gavle, Gavle, Sweden
Wolfgang Kreutzer, University of Canterbury, Christchurch, New Zealand
Marcos A Rodrigues, Sheffield Hallam University, Sheffield ,United Kingdom

MESM’2009

© 2009 EUROSIS-ETI

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

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

All articles published in these Proceedings have been peer reviewed

EUROSIS-ETI Publications are ISI-Thomson and INSPEC referenced

For permission to publish a complete paper write EUROSIS, c/o Philippe Geril, ETI Executive Director, Greenbridge NV, Wetenschapspark 1, Plassendale 1, B-8400 Ostend Belgium

EUROSIS is a Division of ETI Bvba, The European Technology Institute, Torhoutsesteenweg 162, Box 4, B-8400 Ostend, Belgium. (www.eurosis.org)

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

EUROSIS-ETI Publication
ISBN: 978-90-77381-51-9
ISN: 978-90-77381-51-9

Preface

Dear conference delegate,

I have the pleasure to welcome you all to the 10th Middle East Modeling & Simulation Multiconference (MESM2009) Organized by the EUROSIS and hosted by the Lebanese American University in Beirut and Byblos, Lebanon. The MESM2009 is sponsored by the IEEE – UKRI SPC & De Montfort University (UK).

This year's event marks our first visit to Lebanon with the MESM'2009 and we are doubly grateful for the fact that this year's event will be opened by the dean of the Lebanese American University.

The MESM'2009 Conference highlights recent and significant advances in many research areas of modeling and simulation related to Methodology, Simulation Tools, Simulation and AI, Simulation in Electronics and Industrial Simulation..

Your presence at this conference emphasizes an important fact: the challenge and opportunities that surrounding the Modeling and Simulation practice anywhere in the world. We in EUROSIS believe strongly that Modeling & Simulation make a powerful combination to improve the systems and organisations of the 21st century. It will continue to contribute to increase public awareness of the quality of life, and the growing need to improve this through better organizations and systems.

As a General Conference & Programme Chair, I would like to express my thanks to De Montfort University (UK) for giving me the time to chair this conference and thanks also to the committee members for reviewing the papers and to our local chair Dr. Haidar Harmanani, Department of Computer Science, Lebanese American University, Byblos, Lebanon in organizing this event.

Special thanks also to Ahmed Zeki of the International Islamic University Malaysia, Kuala Lumpur, Malaysia for giving a tutorial on the "Introduction to Data Mining and its Applications".

Thanks to my colleague Philippe Geril, executive director of EUROSIS office for supporting the event and for his time. Last but not least thanks to all authors without whom the conference would not be a successful conference.

Professor Dr Marwan Al-Akaidi
General Conference and Programme Chair
EUROSIS – M. East Chair
School of Engineering & Technology,
De Montfort University,
Leicester, LE 1 9BH, UK.
Email: mma@dmu.ac.uk

CONTENTS

Preface ..... **IX**
Scientific Programme **1**
Author Listing **131**

SIMULATION METHODOLOGY AND TOOLS

An Advanced Technique for Rare-Event Simulation Based on Partition of the Region
Amir Jalaly Bidgoly, Mohammad Abdollahi Azgomi and Ali Khalili **5**

Hypertension Induced Remodeling In a Lamellar Model of Arterial Wall
Ahmad Ehsandar, Mohammad Tafazzoli-Shadpour and Mirhamed Sarkarfarshi **11**

Cycle Accurate Simulator Generation Using HARMLESS_
Rola Kassem, Mikaël Briday, Jean-Luc Béchennec, Yvon Trinquet and Guillaume Savaton **18**

SIMULATION AND AI

Dynamization of core models using Description Logics and the ISITS framework
Tobias Lehmann and Andreas Karcher **29**

Constrained Handwriting recognition using Modular Kohonen Neural Networks
Umran Azziz, Jaya Adiguna and Ahmed M. Zeki **35**

Design an Approach for representing Mamdani Fuzzy Rules by Sugeno Rules Depending on Single Perceptron Networks
Rasha E. Karakchi **40**

COMMUNICATION NETWORKS SIMULATION

Design and Implementation of Computer Networks Based on ATM Technology
Saad Alshaban, Lubna K. M. Al-Azzawi and Sattar Sadkhan **49**

Verification of Square Communication grid Protocols via Infinite Petri Nets
T.R. Shmeleva., D.A. Zaitsev and I.D. Zaitsev **53**

CONTENTS

Modeling and Simulation of an Intrusion Tolerant Web Service Architecture Using Coloured Petri Nets
Zahra Aghajani and Mohammad Abdollahi Azgomi **60**

A Cyber Attack Simulation Model for Evaluation of System Availability
Golriz Khazan and Mohammad Abdollahi Azgomi **67**

Video multicast using unequal error protection with Luby Transform codes
Shakeel Ahmad, Raouf Hamzaoui and Marwan Al-Akaidi **72**

COMPLEX SYSTEMS SIMULATION

Towards an Optimization Problem for the Design of Ports
Ghassan El-Chahal, Rafic Younes and Pascal Lafon **79**

Evaluation of the Soft Error Rate of a Space Application Executed by a Real Time Operating System
Paul Peronnard, Esteban Cabanillas, Raoul Velazco, Wassim El Falou and Haissam Ziade **84**

INDUSTRIAL SIMULATION

A Multiagent System for production Synchronization in SME Mechatronics Supply Chain
Jihene Tounsi, Georges Habchi and Julien Boissière **91**

Simulation of dynamic allocation and routing in FMS using potential field
Nadine Zbib, Yves Sallez and Damien Trentesaux **98**

Modelling and Simulation of the Geometry of Cross sectional Variation of Single Yarns
Ionut Neagu, Pascal Bruniaux, Dorin Avram and Adrian Rozenberg (Neagu).. **105**

Modeling of a Pump-less LiBr Absorption System for Residential Consumption
H. Monsef, M. Naghashzadegan, M. Farshad Manesh and S. Roohi **110**

LATE PAPER

RayLang: A Modeling Language for Performance Evaluation of Stochastic Discrete-Event Systems
Ali Khalili and Mohammad Abdollahi Azgomi **119**

SCIENTIFIC PROGRAMME

SIMULATION METHODOLOGY AND TOOLS

AN ADVANCED TECHNIQUE FOR RARE EVENT SIMULATION BASED ON PARTITION OF THE REGION

Amir Jalaly Bidgoly
Mohammad Abdollahi Azgomi
Ali Khalili
Performance and Dependability Engineering Lab., School of Computer Engineering,
Iran University of Science and Technology, Tehran, Iran
E-mail: amir_jalaly@comp.iust.ac.ir, azgomi@iust.ac.ir, al_khalili@comp.iust.ac.ir

KEYWORDS
Rare event simulation, partition of the region (PoR), stochastic discrete-event systems (SDES), stochastic activity networks (SANs).

ABSTRACT

In mission and safety critical applications and some other real areas, the existence of some interested rare events causes the traditional simulation to be inefficient. In these cases, rare event simulation techniques can be used in which the occurrence probability of rare events is increased, usually by utilizing an accelerator. Importance sampling (IS) and importance splitting are two famous techniques for rare event simulation. In this paper we introduce an advanced technique for rare event simulation of stochastic discrete-event systems (SDES) based on partition of the region (PoR) method. Partition of the region is a special kind of stratified sampling variance reduction technique. In a previous paper, we used PoR for simulation of rare events. In this paper, this technique, which is now called *simple PoR*, is improved. The proposed technique, which is called advanced PoR, removes some drawbacks of the simple PoR. To evaluate the efficiency of the new technique, we have done some case studies using stochastic activity networks (SANs). The results of these case studies show a considerable improvement compared to the existing techniques.

INTRODUCTION

Discrete-event simulation is an important method for the modeling and evaluation of systems. In simulation of high-dependable systems, such as switches of ATM networks, and in some applications, such as insurance companies, the existence of *rare events* causes some serious issues. In these applications, to produce accurate results, the time of simulation will be unacceptably long.
Rare events are events with low probability. To solve this problem, a *rare event simulation* technique should be used to increase the occurrence probability of rare events. For this purpose, usually an *accelerator* (Bulhuis et al. 2002; Bucklew 2004; Heegaard 1995; Heidelberger 1995;0 Rubino and Tuffin 2009) is utilized.
Importance sampling and *importance splitting* are two famous methods for rare event simulation. Importance sampling (IS) 0(Blum et al. 1993; Bucklew 2004; Glasserman et al. 1999; Juneja and Shahabuddin 2006; Shahabuddin 1994) as the most famous and important method, alters the probability laws of the model to increase the relative frequency of rare events. The model is then biased, but the bias is compensated by the introduction of a function called the *likelihood ratio*, which is the ratio of the sample path in the original model to the altered model. A major difficulty in applying this method is to figure out how to change the probabilities, in which the IS' estimator has much smaller variance than the original one.
In importance splitting, the probability laws remain unchanged, but an artificial drift toward the rare events is created by terminating with some probability trajectories that seem to go away from it and by splitting (cloning) those that are going in the right direction (L'Ecuyer et al. 2006; Ce'rou et al. 2005; Garvels 2000). This method is presented in (Ville`n-Altamirano and Ville`n-Altamirano 1991; Ville`n-Altamirano and Ville`n-Altamirano 1994) as *REpeative Simulation Trials After Reaching Thresholds (RESTART)* method.
Nevertheless the obvious advantages of rare event simulation, it has not gained enough interests in industrial applications. There exist at least three technical reasons for this problem that is common between almost all rare event simulation techniques (Zimmermann 2008) 0:

- We must have enough background knowledge about the model to apply rare event simulation techniques. An example is the definition of importance function and levels in a splitting setting. Automatic derivation of a good importance function for a given model is still an open research issue (Garvels 2002) 0.
- Rare event simulation algorithms have been developed or adapted for restricted types of models.
- Lack of software tools that are easy to use.

In a previous paper (Jalaly and Azgomi 2008), *partition of the region* (PoR) technique is used for fast simulation of rare events in stochastic discrete-event systems. PoR can be considered as a special kind of *stratified sampling* (SS), which is a well-known variance reduction method. In PoR, the simplicity and efficiency of SS are increased by limiting partitioning of the model into two regions.
In this paper, the technique introduced in (Jalaly and Azgomi 2008) is called *simple PoR* and is improved by introduction of *advanced PoR technique*. The proposed technique eliminates some drawbacks of the simple PoR. Although both simple PoR and advanced PoR can be used for rare event simulation of stochastic discrete-event models, in this paper, stochastic activity networks (SANs) is use in our case studies. The results of simulations show a considerable improvement in comparison to the naïve simulation. The proposed technique is also compared with other existing methods and shows some advantages, including its simplicity and easy usage.

The rest of the paper is organized as follows. In section 2, some preliminaries are presented. The simple PoR technique is introduced in section 3. In section 4, the advanced PoR technique is presented. Finally, the paper ends with conclusions in section 5.

PRELIMINARIES

Consider a rare event $\varepsilon \subset \Omega$ where Ω is the state space of a model with the probability measure P. Our interest is to estimate the probability $P(\varepsilon)$. Let's define $I(\varepsilon)$ as an indicator function of the event ε which is equal to 1, if outcomes belong to ε; otherwise, it is equal to 0. Let's γ denotes the probability $P(\varepsilon)$. In Monte Carlo method, this can be estimated by generating n independent samples ($I_1(\varepsilon)$, $I_2(\varepsilon)$, ..., $I_n(\varepsilon)$) of $I(\varepsilon)$ via naïve simulation and taking the average $\frac{1}{n}\sum_{i=1}^{n} I_i(\varepsilon)$ as an estimator of γ. Let us use the symbol γ_n for this estimator. In rare event simulation, n would have to be quite large to estimate γ reliably, because most of the samples of $I(\varepsilon)$ would be zero. The central limit theorem proves useful in developing a confidence interval (CI) for estimation and could be used to determine the n necessary for accurate estimation (Juneja and Shahabuddin 2006). For large n, an approximate (1-α)100% CI for γ is given by:

$$\gamma_n(P) \pm z_{\alpha/2}\sqrt{\frac{\sigma_P^2(I(\varepsilon))}{n}} \tag{1}$$

where $\sigma^2(I(\varepsilon))$ denotes the variance of $I(\varepsilon)$ and z_x is a number which satisfies the relation $P(N(0, 1) \geq z_x) = x$. Here, $N(0,1)$ denotes a normally distributed random variable with mean zero and variance one. On the other hand, variance of $I(\varepsilon)$ is equal to $(1-\gamma)\gamma$. According to the above equation, for a reliable estimator, n must be chosen to make the following relative confidence interval (Eq. 2) sufficiency small (Heidelberger 1995):

$$\frac{CI}{\gamma} = z_{\alpha/2}\sqrt{\frac{(1-\gamma)}{n\gamma}} \tag{2}$$

This implies that as $\gamma \to 0$, $n \to \infty$ to obtain a reasonable level of relative accuracy. Thus, naïve simulation becomes an infeasible proposition for sufficiently rare events. In the context of rare event simulation, the asymptotic robustness of an estimator is qualified with some properties, which is also called asymptotic optimality. *Bounded relative error* (BRE) and *logarithmic efficiency* (LE) are two most important properties in this regard (L'Ecuyer et al 2009; Tuffin 1999; Tuffin 2004).

In equation (1), less sample size can be used by reducing variance and thus, variance reduction is the primary approach for rare event simulation.

In the remainder of this section, a variance reduction technique called *partition of the region* (PoR) will be described. Then, stochastic activity networks (SANs), which is used in our case studies, will be briefly introduced.

Partition of the Region

Again, consider a rare event $\varepsilon \subset \Omega$, where Ω is the set of state spaces with the probability measure P. Our interest is to estimate the probability $P(\varepsilon)$ in D, where D is the region used for generating samples of $I(\varepsilon)$. This probability can be considered as the following equation:

$$P(\varepsilon) = E[I(\varepsilon)] = \int_D I(\varepsilon)dP \tag{3}$$

In partition of the region (PoR) (Rubinstein 1981) technique, the region D is broken into two parts, D_1 and D_2, such that $D=D_1 \cup D_2$, representing the integral (3) as follows:

$$P(\varepsilon) = \int_D I(\varepsilon)dP = \int_{D_1} I(\varepsilon)dP + \int_{D_2`} I(\varepsilon)dP \tag{4}$$

Assume the second integral lacks any rare event and can be estimated simply by a traditional estimator. In addition, consider a probability P' as the sum of the probability of all states ω where $\omega \in \Omega$ and $\omega \subset D_2$. This probability is the *likelihood ratio* and gives an unbiased estimator for the simulator. Let's rewrite equation (4) using P', as follows:

$$\begin{aligned} P(\varepsilon) = \; & P' \int_{D_1} I(\varepsilon)dP + (1-P') \int_{D_2`} I(\varepsilon)dP = \\ & P'\gamma_2 + (1-P')\sum_{i=1}^{n} .I_i(\varepsilon) \end{aligned} \tag{5}$$

Here, γ_2 denotes this integral:

$$\gamma_2 = \int_{D_2} I(\varepsilon)dP \tag{6}$$

In practice, D_2 will be defined in a way that does not have any rare event (i.e. $I(\varepsilon)=0$). Therefore, in simulation, there is not needed to simulate this part.

In PoR technique, D_1 and D_2 must be independent. This causes some limitations in implementation of PoR. Due to this limitation we have implemented PoR in two levels.

- In the first level, as introduced in (Jalaly and Azgomi 2008) and is called simple PoR, can be used with a wide range of models, but not all models.
- In the second level and to apply PoR to all kinds of stochastic discrete-event models, we have integrated PoR with the IS method. The resulting method it is called *advanced PoR*.

Stochastic Activity Networks

Stochastic activity networks (SANs) (Sanders and Meyer 2001), as a general and stochastic extension of the Petri nets, are powerful and flexible models for concurrent and distributed systems. These models are supported by several powerful modeling tools, such as *UltraSAN* (Sanders et al. 1995), *Möbius* (Deavours et al. 2002) and *PDETool* (Khalili et al. 2009).

Elements of SANs are *places, gates* and *activities*. Places are same as Petri nets. Gates are used to connect places and activities and have two types: *input* and *output*. Input gates connect one or more places to a single activity and have a *predicate* and a *function*. Output gates connect an activity to one or more places and have only a function. The other element of SANs is the activity. There are two types of activities: *timed* and *instantaneous*. Timed activities have a delay to *complete*. Delay time is represented by a distribution function called *activity time distribution. F*(.,μ; *a*) denotes the distribution function for activity *a* in marking μ. Activities also have cases in their outputs to show uncertainly about action taken at completing. *C*(.,μ; *a*) used for showing case distribution function of activity *a* in marking μ.

An activity is called enabled when all gates and places connected directly to the activity hold true, i.e. the predicates of gates must return true and places must have at least one token. When an activity is enabled, it awaits completion. Time for delay taken from time activity distribution called *activity time.* After completion, first the input functions are executed and then the output functions. An enabled activity does not require completion. During activity time, SAN can move to a marking that activity be no more enabled and *aborted.*

For more information about SANs, including their formal definition and behavior, please see (Sanders and Meyer 2001).

THE SIMPLE POR TECHNIQUE

In this section, we present an overview of the simple PoR technique (Jalaly and Azgomi 2008). Consider a stochastic discrete-event model with state space Ω and transition set Π. The aim is estimating the probability of $P(\varepsilon)$ where $\varepsilon \subset \Omega$ is the set of rare events and $P(\varepsilon)\to 0$. Let's define an indicator function on state space as follows:

$$I' : \Omega \to \{0,1\}$$

This indicator function is called *partitioner function*. Using the partitioner function, the state space of the model can be partitioned into two separate regions: Ω_1 and Ω_2, which are corresponding to D_1 and D_2 in equation (4), respectively.

$$\Omega_1 = \{\omega \in \Omega \mid I'(\omega) = 1\}$$

$$\Omega_2 = \{\omega \in \Omega \mid I'(\omega) = 0\}$$

The partitioner must be chosen to divide the model in which the first part includes all rare events (i.e. ε) and the second is free of them. In practice, we need to have enough knowledge about the model to find a good partitioner function. Different models need different partitioners. For example, if the aim of simulation is to find the blocking probability of an M/M/1/20 queue, the partitioner may divide the model into two parts as follows: (1) the first part contains those states that have less than 15 customers in the queue and (2) the second part contains the states with 15 customers or more.

To evaluate unavailability in a high-dependable system we may need another kind of partitioning. Decreasing the state space increases the probability of rare events. On the other hand, the omitted region lacks any rare event. Therefore, there is no need to simulate this part of the model.

The next step in simple PoR is computing the probability P' used in equation (5), which can be estimated by the following indicator function:

$$I(\omega) = \begin{cases} 1 & \omega \in \Omega_2 \\ 0 & \omega \in \Omega_1 \end{cases} \tag{7}$$

Now, the following equation is valid in evaluation of ε:

$$\left.\begin{aligned} P(\varepsilon) &= \int_{\Omega} I(\varepsilon)dP \\ &= (1-P')\int_{\Omega_1} I(\varepsilon)dP \\ &\quad + P'\int_{\Omega_2} I(\varepsilon)dP \\ \int_{\Omega_2} I(\varepsilon)dP &= 0 \end{aligned}\right\} \Rightarrow P(\varepsilon) = (1-P')\int_{\Omega_1} I(\varepsilon)dP \tag{8}$$

Based on equation (8), the following estimator can be considered as the estimator of the model:

$$\hat{\gamma}_n(P) = \frac{1}{n}\sum I(\varepsilon).(1-P') \tag{9}$$

This estimator would take its samples only from the first region.

PoR turns out to a reduction in variance. Note that the variance of $I(\varepsilon)$ is equal to $(1-\gamma)\gamma$ where γ denotes the probability $P(\varepsilon)$. Thus, we have:

$$\frac{\sigma^2_{PoR}(\gamma)}{\sigma^2(\gamma)} = \frac{\sigma^2\left((1-P')\gamma\right)}{\sigma^2(\gamma)} = (1-P') \tag{10}$$

Hence, guaranteed variance reduction by a factor of $(1-P')$ is achieved.

In the remainder of this section, we study applying simple PoR technique on a SAN model and discuss about the simple PoR advantages and disadvantage.

A Case Study Using Simple PoR

The example examined in this section is an M/M/1 queue. The SAN model corresponding to an M/M/1 queue is shown in Figure 1. In this model, timed activity *Job_Income* models arrival process and *Process* models the server. There is a place, named *Queue*, which models the queue waiting line. Interarrival and service times are exponentially distributed with rate 1.0 and 2.0, respectively. The aim of simulation is to compute the probability of having more than *n* customers in the queue. The desired probability, for n=20, is first reached through analytical solution that is equal to 4.7e-7. For applying the simple POR technique, the following indicator function is defined as the partitioner function:

$$I' = \begin{cases} 1 & Queue.Mark > 12 \\ 0 & otherwise \end{cases}$$

This partitioner forces the model to continue the simulation while at least 12 tokens exist in the place *Queue*. This leads to an increase in the probability of existence of more than 20 tokens in this place, which in turn accelerates reaching the result. To apply this partitioning, we have used an input gate *ig1* for the activity *Process*, which its predicate and function are shown in Table 1.

Table 1: *ig1* Input Gate Properties

Predicate	Function
If (*Queue.Mark*>12) *return true;* *else* *return false;*	*Queue.Mark--;*

To evaluate the model, we have used PDETool (Khalili et al. 2009). The tool is run on an Intel 1.8 GHz with 1 GB of RAM. Because, the simulation using POR takes less than 100 seconds, the maximum time for naïve simulation is considered as 10000 seconds. The simulation results with confidence level 99% and confidence interval 10% for naïve simulation in comparison to the PoR-based technique are presented in Table 2. This comparison is done in three different values of *n*. The simulation has been executed for *n*=20, 30 and 45 (i.e. probability of being more than 20, 30, and 45 tokens in the queue) in which the partitioner function is changed according to the values of *n* (i.e. it forces the model to run in the states more than 12, 20, and 30 tokens).

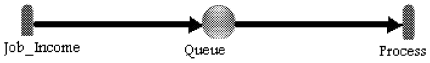


Figure 1: SAN Model of an M/M/1 Queue

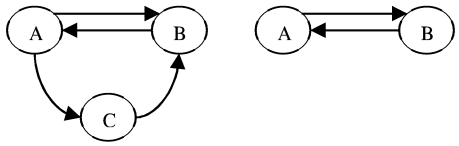
Table 2: Naïve Simulation vs. Simple PoR for M/M/1 Queue

	Analytical Solver	Naïve Simulation		Simple PoR	
		Time	Results	Time	Results
P(>20)	4.768e-7	2856	4.598e-7	61	4.758e-7
P(>30)	3.725e-9	>>10000	--	72	3.298e-9
P(>45)	1.455e-11	>>10000	--	110	1.030e-11

Advantages and Limitations of Simple PoR

In comparison with other techniques for rare event simulation, simple PoR technique has some important advantages, including its simplicity and its power. In addition, it does not require any special dedicated simulator and can be used by the existing simulators (ex. Möbius). To do so, the process of rare event simulation must be done within two steps. First, simulating the model to estimate P' as the likelihood ratio and then, simulating the modified model (obtained by the partitioner function) and correcting the results using the likelihood ratio in the second step. As most of the simulation tools do not support rare event simulation, this method can simply be used for this purpose. Moreover, becuase the simple PoR does not require much information about the evaluated model (as importance sampling needs), it will be easier to set up a meaningful simulation study.

The simple PoR method has a major limitation as well. This method is applicable only when the omitted states and remaining states (i.e. the first and the second region) are independent. For example, Figure 2 (a) shows a simple CTMC model with three states in which all output transitions for each state have the rate 1. Using the simple PoR method for this model causes the state *C* to be omitted and we reach a new modified model as depicted in Figure 2 (b).



(a) (b)

Figure 2: A Simple Model with 3 States, (a) before and (b) after Using Simple PoR Method

In Figure 2 (a), the steady-state probability of state *C* is 0.25. Thus, P'=0.25 as the likelihood ratio. Based on simple PoR method, the probability system being in the state *A* in the modified model using the likelihood ratio must be equal to its probability in the original model. This is while we have:

$$\begin{Bmatrix} P_{PoR}(A) = (1-P') * P(A) = 0.75 * 0.5 = 0.375 \\ P_{Org}(A) = 0.25 \end{Bmatrix} \Rightarrow P_{PoR}(A) \neq P_{Org}(A) \quad (11)$$

$P_{Org}(A)$ and $P_{PoR}(A)$ are the steady state probability of state *A* in the original and modified models, respectively. This example clearly shows that simple PoR method could not be used with all models.

In the next section, this problem will be solved by a modified method, which integrates PoR and IS technique and is called *advanced PoR*.

ADVANCED POR TECHNIQUE

As mentioned in the previous section, while the simple PoR method is powerful, it could not be used with all models. It assumes that after omitting some states, the probability of the other states increases with the same order, which is not true in most models. As an example, the model shown in Figure 2 is opposite to this assumption. In this model, after removing state *C*, the steady state probability of state *A* increases with ratio 2 (i.e. *P(A)* is equal to 0.25 in the original model and 0.5 in the modified model) while the steady state probability of state *B* is unchanged (*P(B)* is equal to 0.5 in both models). This problem can be solved by introducing a new method for computing the likelihood ratio. Here, the main assumption is that the likelihood ratio may not be constant during the simulation progress.

In this section, a second technique called *advanced PoR* is introduced that uses the IS method for this purpose. Consider a stochastic model with state space Ω, transition set Π and a rare event ε, as presented in the previous section. Suppose we have two separate regions Ω_1 and Ω_2 ($\Omega_1 \cup \Omega_2 = \Omega$) obtained by defining a partitioner function ε'. Let's divide the transition set into the following two subsets:

$$\forall \pi \in \Pi, \omega_i, \omega_j \in \Omega: \omega_i \xrightarrow{\pi} \omega_j \begin{cases} \pi \in \Pi_2 & \omega_i \in \Omega_1, \omega_j \in \Omega_2 \\ \pi \in \Pi_1 & otherwise \end{cases} \quad \Pi_1 \cup \Pi_2 = \Pi \quad (12)$$

Generally, it is not compulsory to divide events by a partitioner defined in simple PoR. To extend the method, we define a new partitioner function called *advanced PoR partitioner*, as follows:

$$I': \Pi \to \{0,1\} \qquad \forall \pi \in \Pi: \begin{cases} \pi \in \Pi_1 & I'(\pi) = 1 \\ \pi \in \Pi_2 & I'(\pi) = 0 \end{cases} \quad (13)$$

Assume $f(.,\pi)$ and $f^*(.,\pi)$ denote the probability density functions of the transition π in the original and modified model respectively. Let's define a new distribution P^* for the model in which the rate of each transition is modified as follows.

$$\forall \pi \in \Pi: f^*(.,\pi) = \begin{cases} f(.,\pi) & \pi \in \Pi_1 \\ \mu: \mu \to 0 & \pi \in \Pi_2 \end{cases} \quad (14)$$

Here, $f^*(.,\pi) = \mu: \mu \to 0$ means $f^*(.,\pi)$ converge to zero that cause transition π fires only in infinity (e.g. in exponential distribution with rate λ when $\lambda \to 0$). In practice, the transitions, which have such a probability measure, will never fire. Defining this distribution, the model can be considered as shown in Figure 3. Regarding the region containing the initial state of the model, we can claim that the Ω_2 is eliminated from the model or at least its probability is decreased to an extremely small value.

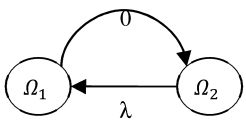


Figure 3: Applying Advanced PoR on a Model

Applying the partitioner function with the above probability measure makes the simulator capable of evaluate the likelihood ratio in each sample path based on IS' principles. Let's denote P^* the new probability distribution. The following equation shows the basic principle in this regard:

$$P(\varepsilon) = \int I(\varepsilon) dP = \int I(\varepsilon) \frac{dP^*}{dP^*} dP = \int I(\varepsilon) \frac{dP}{dP^*} dP^* \quad (15)$$

$L=dP/dP^*$, the quotient of probability measures, is called the *importance sampling likelihood ratio*. To estimate $I(\varepsilon)$ under probability measure P^*, we have to compensate the bias by actually simulating $I(\varepsilon)L$. An estimator of $E_{P^*}(I(\varepsilon))$ is then equal to:

$$\frac{1}{n} \sum_{i=1}^{n} I_i(\varepsilon) L_i \quad (16)$$

For more details about IS and likelihood ratio computation, refer to (Juneja and Shahabuddin 2006; Nicola et al. 1993; Nicola et al. 2001; Obal 1993; Obal and Sanders 1994).

A Case Study Using Advanced PoR

In this section, the simulation results of a sample SAN model using advanced PoR technique are presented. The model shown in Figure 4 is a machine repairman system that uses the group repair policy. There are two types of components in the system with same failure rates. The places labeled as *type_1* and *type_2* are used to model the two types of working components. In this case, there are 15 components of each type. The timed activities *fail_*1 and *fail_*2 model the failures for each component. The markings of the places *failed_*1 and *failed_*2 represent the number of components of each type that have failed. There are two repairmen in the system. The repairmen begin to repair when at least one component is failed, and when the repair is completed, all components of that type will be as good as new one. Repairing type one components is given preemptive priority over repairing type two components. This repair policy is implemented in the input gate *policy*. All activities are exponential with rate 1.0. If all components of both types fail, the system fails, and all repair activity halts; the failed state is an absorbing state. The aim of simulation is to compute the unreliability during the interval [0, 100]. Due to the large number of redundant components and group repair policy, this property is very small and can be considered as a rare event.

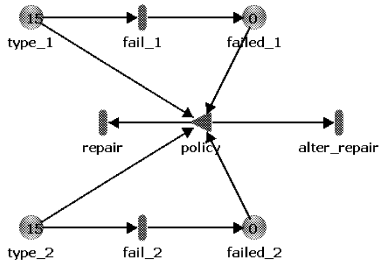


Figure 4: A Sample Machine Repairman Model

For applying advanced PoR, the following probability density function is chosen for *fail_*1 and *fail_*2 activities, which means that no components will be repaired if at least 7 components of each type have failed:

$$f(.,fail) = \begin{cases} \exp(\lambda) \,|\lambda \to 0 & Place4.Mark > 7 \\ & and\ Place6.Mark > 7 \\ \exp(1) & otherwise \end{cases}$$

Therefore, the unreliability of the model will be increased and there will be more chance to run in the failed state.

Table 3 shows the simulation results using advanced PoR technique and the naïve simulation. The maximum time for naïve simulation is considered as 50000 seconds. Simulations run with relative half-widths of 95% confidence intervals using PDETool. For comparison purposes, the model is also solved numerically using the transient solver of Möbius (Deavours et al. 2002).

Table 3: Simulation Results Using Advanced POR vs. Naïve Simulation for the Model of Figure 4

	Analytical Solver	Naïve Simulation	Advanced PoR
Result	1.00e-007	--	1.09e-7
Time	--	>>50000	1536
Error	--	--	9%

Advanced PoR vs. Importance Sampling

Although IS and advanced PoR both simulate and run in a same way using likelihood ratio computation, they have different effects on the model. IS method accelerates rare event occurrence by changing probability law. This can be done by increasing their distribution rate. In this approach, the whole model must be simulated and no part is removed from the model. On the other hand, the advanced PoR technique decreases the probability of other events to zero, instead of increasing the probability of rare events, so causes they never occur in simulation. As mentioned before, because of the absence of competitive events besides rare ones, the probability of their occurrence increases.

In other point of view, the most important problem in IS is choosing a good distribution for rare events that makes variance least. Some choices may lead to large and even infinite variance. In practice, as on a small set in ε, the likelihood ratio may take large values leading to a blow-up in the variance of the estimator (Juneja and Shahabuddin 2006). Thus, it is required to closely study the characteristics of a model to choose a good distribution. On the other hand, finding a part of the model for removing does not require as much information as about the evaluated model as finding how the rare events probability must be increased. In most cases, repair events or other competitive events are a good option to remove. Therefore, it is easier to set up a meaningful simulation study using advanced PoR. As an example, we also simulate the model represented in Figure 4 using IS technique. For this purpose, the rate of fail events are increased to 4 and the model is simulated again using the likelihood ratio. The results are shown in Table 4, which is also compared with advanced PoR. Here, the maximum time for simulation is considered as 10000 seconds.

Table 4: Simulation Results of the Model Shown in Figure 4 Using Advanced POR and IS

	Importance Sampling	Advanced PoR
Result	--	1.09e-7
Time	>>10000	1536
Error	--	9%

We remind that although it may be possible to find a distribution for IS that leads to the results as well as the results using advanced PoR or even better in some cases, but

it requires more information about the model and its characteristics.

CONCLUSIONS

In this paper, an advance technique for rare event simulation of stochastic discrete-event systems is presented that is based on the partition of the region technique. This method makes rare events to occur more frequently by omitting other competitive events. Because, simple PoR suffers a major limitation (i.e. it cannot be applied on all kinds of models), advanced PoR technique is proposed. Advanced PoR can be used for all kinds of models and improves the simple PoR using IS principles. In this paper, both methods are briefly compared with another famous technique for rare event simulation (i.e. IS). The advanced PoR technique is capable of be applied on any stochastic discrete-event model. However, in this paper we used stochastic activity networks (SANs) in our case studies. The results show a meaningful improvement in simulation time. Furthermore, this technique can simply be implemented for other discrete-event models, such as stochastic extensions of Petri nets.

REFRENCES

Blum, A. M; et al. 1993. "System Availability Estimator (SAVE) Language Reference and User Manual." Version 4.0, Tech. Rep. RA219S, IBM Thomas J. Watson Research Center.

Bulhuis, et al. 2002. "Transition Path Sampling: Throwing Robe Over Rough Mountain Passes, in the Dark." *Annual Review on Physical Chemistry*, Vol. 53, 291-318.

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

Ce'rou, F.; et al. 2005. "Limit Theorems for the Multilevel Splitting Algorithm in the Simulation of Rare Events." In *Proc. of the 2005 Winter Simulation Conf.*, 682-691.

Deavours D. D. and et al. 2002. "The Möbius Framework and Its Implementation." *IEEE Trans. on Soft. Eng.*, 956-969.

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

Jalaly Bidgoly, A. and M. Abdollahi Azgomi. 2008. "Rare Event Simulation of Stochastic Activity Networks Using Partition of the Region Technique." In *Proc. of the 24th UK Performance Engineering Workshop (UKPEW'08)*, Imperial College London, UK, 107-122.

Garvels, M. J.; et al. 2002. "On the Importance Function in Splitting Simulation." *European Trans. on Telecom.*, vol. 13, no. 4, 363-371.

Glasserman, P.; et al. 1999. "Asymptotically Optimal Importance Sampling and Stratification for Pricing Path Dependent Options." *Mathematical Finance,* vol. 9 , no. 2, 117-152.

Heegaard, P.E. 1995. "Speed-Up Techniques for Simulation." *Telektronikk.*

Heidelberger, P. 1995. "Fast Simulation of Rare Events in Queueing and Reliability Models." *ACM Trans. on Modeling and Comp. Sim.*, vol. 5, no. 1, 43-85.

Juneja, S. and P. Shahabuddin. 2006. "*Rare-Event Simulation Techniques: An Introduction and Recent Advances.*" Amsterdam, The Netherlands: Elsevier, Handbooks in OR and Management Science.

Khalili, A.; A. Jalaly Bidgoly and M. Abdollahi Azgomi. 2009. "PDETool: A Multi-formalism Modeling Tool for Discrete-Event Systems Based on SDES Description." *LNCS*, vol. 5606, 343-352.

L'Ecuyer, P.; et al. 2006. "Splitting for Rare-Event Simulation." In *Proc. of the 2006 Winter Sim. Conf.*, 137-148.

L'Ecuyer, P.; et al. 2009. "Asymptotic Robustness of Estimators in Rare-Event Simulation." *ACM Trans. on Modeling and Comp. Sim.*, to appear.

Nicola, V. F.; et al. 1993. "Fast Simulation of Highly Dependable Systems with General Failure and Repair Processes." *IEEE Trans. on Comp.*, 1440-1452.

Nicola, V.F.; et al. 2001. "Techniques for Fast Simulation of Models of Highly Dependable Systems." *IEEE Trans.on Reliability*, vol. 50, no. 3, 246-264.

Obal II, W.D. 1993. *Importance Sampling Based of SPN-Based Reward Models*. Master Thesis, The University of Arizona.

Obal II, W. D. and W. H. Sanders. 1994. "Importance Sampling Simulation in UltraSAN." *Simulation*, vol. 64, no. 2, 98-111.

Rubinstein, R.Y. 1981. *Simulation and Monte Carlo Method*. USA: John Wiley & Sons Inc.

Rubino, G. and B. Tuffin. 2009. *Rare-Event Simulation Using Monte Carlo Methods*. John Wiley.

Sanders, W.H. et al. 1995. "The UltraSAN Modeling Environment." *Performance Evaluation*, vol. 24, 1-33.

Sanders, W. H. and J. F. Meyer. 2001. "Stochastic Activity Networks: Formal Definitions and Concepts." LNCS, vol. 2090, Springer-Verlag, 315-343.

Shahabuddin, P. 1994. "Importance Sampling for the Simulation of Highly Reliable Markovian Systems." *Management Science*, vol. 41, no. 3, pp. 333-352.

Shreider, Yu. A. 1966. *The Monte Carlo Method: The Method of Statistical Trials*. Pergamum, Elmsford, New York.

Tuffin, B. 1999. "Bounded Normal Approximation in Simulations of Highly Reliable Markovian Systems." *J. of Applied Probability*, vol. 36, no. 4, 974-986.

Tuffin, B. 2004. "On Numerical Problems in Simulations of Highly Reliable Markovian Systems." In *Proc. of the 1st Int'l Conf. on Quantitative Evaluation of SysTems (QEST)*, University of Twente, Enschede, The Netherlands, 156–164.

Ville`n-Altamirano M. and J. Ville`n-Altamirano. 1991. "RESTART: A Method for Accelerating Rare Event Simulations." *Queueing Perf. and Control in ATM*, 71-76.

Ville`n-Altamirano M. and J. Ville`n-Altamirano. 1994. "RESTART: A Straightforward Method for Fast Simulation of Rare events." In *Pro. of the 1994 Winter Sim. Conf.*, 282-289.

Ville`n-Altamirano M. and J. Ville`n-Altamirano. 2006. "On the Efficiency of RESTART for Multidimensional Systems." *ACM Trans. on Modeling and Comp. Sim.*, vol. 16, 251-279.

Zimmermann, A. 2008. "RESTART Simulation of Colored Stochastic Petri Nets." in *7th Int. Workshop on Rare Event Simulation (RESIM'08)*, Rennes, France, 143-152.

BIOGRAPHY

AMIR JALALY BIDGOLY received the B.Sc. degree in computer engineering (software) from Kashan University (2005) and M.Sc. degree in software engineering from Iran University of Science and Technology (July 2009). Title of his M.Sc. thesis was "New Techniques for Fast Simulation of Stochastic Petri Nets," which has been done under supervision of Dr. Mohammad Abdollahi Azgomi.

Mr. Jalaly's research interests include modeling techniques and simulation tools. He has published several papers in international conferences.

Hypertension Induced Remodeling In a Lamellar Model of Arterial Wall

Ahmad Ehsandar
Cardiovascular Engineering Lab
Biomedical Engineering Department
Amirkabir University of Technology
Tehran, Iran
ahmad.ehsandar@gmail.com

Mohammad Tafazzoli-Shadpour
Cardiovascular Engineering Lab
Faculty of Biomedical Engineering
Amirkabir University of Technology
Tehran, Iran
tafazoli@aut.ac.ir

Mirhamed Sarkarfarshi
Mechanical Engineering Department
University of Tehran
Tehran, Iran
sarkarfarshi@gmail.com

KEYWORDS
Arterial wall, Remodeling, Lamellae, Hypertension, Finite Element Method, Neural Network, Genetic Algorithm

ABSTRACT

The process of remodeling in arterial wall was studied on the basis of finite element method on a lamellar model of arterial wall. For simulating sustained hypertension, internal blood pressure was increased step by step from normotensive to hypertensive conditions. Arterial remodeling was modeled with a numerical method based on the deviation of the wall circumferential stress of the lamellae in hypertensive and normotensive conditions. Geometrical and mechanical properties of the models were allocated based on human thoracic and abdominal aortas. Arterial wall was considered to be lamellar thick-walled composed of nonlinear elastic incompressible material to determine the effect of remodeling at each site of the artery independently. The total number of layers across the thickness of media for abdominal aortas were 28 and 29 respectively according to reported human data. Results showed that hypertension caused the wall thickening especially in sub-intimal region to maintain the circumferential contour constant across the thickness. Results showed that artery's inner layers of artery thicken more that outer layers in response to hypertension. Also, at the same hypertensive pressure, remodeling rate in abdominal aorta was greater than thoracic aorta. The model predictions were in good agreement with published experimental results.

INTRODUCTION

Living tissues are capable of responding to external stimuli by alteration in their structure and mechanical properties. In the arterial system, remodeling is generally an adaptive response to sustained changes in blood pressure or flow to restore or optimize functional characteristics of the artery [Han and Ku 2001; Jacobsen et al. 2003]. Critical levels of external stimuli may cause arterial wall damage or degeneration due to lack of capability of optimized remodeling [Sho et al. 2004; Ghiadoni et al. 1998; Taber 1998]. Excessive circumferential stress across the arterial wall induced by altered blood pressure may lead to pathological conditions such as endothelial damage, and consequently artherogenesis and plaque formation. Pulsatile blood pressure and flow are major determinants of wall remodeling. Studies have shown that elevation of blood pressure and change in pressure pulse lead to alteration of wall geometry [Pries et al. 2005; Boutouyrie et al. 2003]. Variation of blood flow also remodels arterial wall [Michael et al. 2000]. The combination of both parameters, i.e. blood pressure and flow, has also been studied through fluid-structure interaction modeling. It has been shown that elevated blood flow leads to increased internal diameter in specific arteries [Masuda et al. 1999; Qiu and Tarbell 2000; Sho et al. 2001]; and increased blood pressure results in elevation of the thickness to diameter ratio in some arteries[Hashimoto et al. 1999; Nichols and O'Rourke 1998; Taber 1998]. The adaptive response of arterial wall to alterations in blood pressure and flow is through two stresses: arterial wall circumferential stress induced by blood pressure, and wall shear stress caused by blood flow [Jacobsen et al. 2003; Davies 1995; Langille et al. 1989]. Different patterns of remodeling are observed among arterial sites through variations of pressure and flow [Carretero 2005]. Arterial remodeling is categorized based on two geometrical parameters: diameter and wall cross sectional area. It might be in forms of internal (reduction in internal diameter) or external (increase in internal diameter). Considering cross sectional area, arterial remodeling is categorized as hypotrophic (reduction in wall cross sectional area), eutrophic (constant wall cross sectional area), and hypertrophic (increase in wall cross sectional area) [Mulvany et al. 1996]. Small arteries undergo either inward eutrophic or hypertrophic remodeling [Hope et al. 2001]. Hypertension, state of persistent elevation of blood pressure, is a major risk factor of many cardiovascular diseases. Arterial wall is capable of remodeling to maintain its functions under increased blood pressure and the resultant elevated circumferential stress. Such remodeling is optimized in a specific range of wall circumferential stress in different arterial sites. The optimum remodeling pattern defines the geometrical variation in different arteries by alteration in the ratio of thickness to diameter in the optimized range of circumferential stress [Han and Ku 2001; Ghiadoni et al. 1998; Jacobsen et al. 2003]. The elevated blood pressure leads to increased circumferential stress. Since maximum circumferential stress occurs on the inner wall, the endothelial layer is exposed to the elevated strain for the hypertensive arteries. This might lead to endothelial damage which is a key risk factor in arterial degeneration [Taber 1998; Fung 1988]. The pressure induced remodeling causes decrease in the wall circumferential stress to stabilize arterial function and decrease possibility of endothelial damage. In critical situations when arterial wall is not capable of adequate remodeling, the excessive circumferential stress contributes to wall degeneration. Mathematical and numerical models have been developed to describe arterial wall remodeling in such a way that the tensile stress distributions across the arterial wall remains constant. Rachev et al. used a model of thick-walled tube

with a non-linear elastic compressible material [Rachev et al. 1996a, Rachev 1996b]. Effects of hypertension were studied by stepwise increases in blood pressure on a model of zero stress state of the artery, and deviations from normotensive circumferential stress were studied. Such model has been expanded by Rachev to a thick-walled two-layer cylindrical model made of nonlinear elastic incompressible material. Such remodeling algorithms include change in geometry of the arterial wall. To include alteration of material property by remodeling, a phenomenological strain energy function was included as an indicator of collagen engagement probability density function, although no clear physiological meaning was attributed for such function [Tsamis and Stergiopulos 2007]. Growth law has been used as a remodeling theory based on stretch ratios depending linearly on smooth muscle fiber stress and on the shear stress due to blood flow on the endothelium [Ghiadoni et al. 1998]. The model included a thick-walled tube composed of two pseudoelastic, orthotropic layers representing the media and the adventitia. The rate of volumetric growth depended linearly on the local smooth muscle fibers and the blood flow induced shear stress on the endothelium. The model was developed by inclusion of residual stress of the arterial wall [Taber 2001]. Liu and Fung have studied the relationship between hypertension, hypertrophy, and opening angle of zero-stress state of arteries following aortic constriction of rat aorta. Their results showed a constant inner radius after constriction describing no significant change in blood flow and wall shear stress. The wall thickness increased rapidly in the first few days after the onset of hypertension and then gradually attained a new homeostatic value [Liu and Fung 1989]. Calculated mean circumferential stress at the homeostatic state is equal to the stress of the normotensive aorta, indicating that the vessel thickens to restore the mean circumferential stress. It has been suggested that adaptation based on mechanical properties is slower than geometrical based [Matsumoto and Hayashi 1994]. Previous studies have used a uniform homogenous structure for the arterial media which is the major contributor to the mechanics of the wall. Arterial wall consists of lamellae and inter-lamellar layers with different material properties [Wolinsky and Glagov 1969; Wolinsky and Glagov 1967; Clark and Glagov 1985]. Such layers are uniformly distributed across the wall thickness [Dobrin 1999]. The lamellar structure necessitates a composite like structure in which the geometry of the layers is in direct correlation with the material property of wall components. The inter-lamellae are significantly stiffer than the lamellae due to the high value of collagenous fibers. Hence, by elevation of inter-lamellar thickness, arterial wall becomes stiffer. Such correlation makes that possible to build a remodeling algorithm considering both geometry and mechanical properties. Since previous studies have considered the arterial media as one layer, the remodeling algorithm was considered either purely based on the geometry, or mechanical behavior. Taking into account the lamellar structure of the wall, both geometric and mechanical adaptations become co-dependent. By alteration of geometry, mechanical properties change accordingly with an orthotropic and non-linear algorithm [Holzapfel and Ogden 2003]. In this study, arterial wall remodeling induced by hypertension was investigated based on the finite element numerical stress analysis in a lamellar structure comparable to the geometry of abdominal aortas [Wolinsky and Glagov 1969]. Thus, the intrinsic nonlinearity in mechanical properties of the arterial wall has been significantly resembled in such multi-layer modelling approach [Nakashima and Tanikawa 1971].

METHODS AND MATERIALS

This study aims in calculation of remodeled structure of a lamellar model of hypertensive arterial wall and the thickness of layers by means of optimization process in such way that the circumferential stress distribution across the wall remains constant. To simulate the remodeling process, internal pressure as the mean arterial pressure was altered gradually from a normotensive value of 100 mmHg to a hypertensive value of 160 mmHg with step size of 10mmHg. Analyzing different geometries established with the change of layers thickness, those which had the closest stress result to normotensive determined as nearest solution to remodeled configuration. To avoid excessive required CPU time for running the analysis, for each step between 30 to 36 different geometries were analyzed and results of the induced circumferential stress were compared to normotensive configuration of layers with the pressure of 100mmHG. The optimum configuration of layers thickness which was the nearest stress answer to normotensive stress, was determined by means of Artificial Neural Network and Genetic Algorithm method to minimum each layer's circumferential stress difference with normotensive circumferential stress.

Mechanical Properties

The mechanical and geometrical properties of the arterial wall are based on experimental investigations on young human abdominal aortas with normotensive blood pressure [Michael et al. 2000]. The composite structure of the model is assumed to be incompressible due to high content of water in soft tissues, and effects of viscoelasticity is considered negligible in the given conditions [Masuda et al. 1999]

The Young's modules shown in the table 1 are extracted from the stress-strain plot (pressure-diameter) in the arterial wall. The geometrical values of abdominal aorta have been reported in depressurized condition.

Unlike previous studies, a lamellar model is introduced for the artery. A lamellar model helps simulating mechanical behavior of the artery at its best resemblance to the real arterial mechanical properties; moreover, it helps determining the independent behavior of each layer in the remodeling process of the whole arterial wall. As shown in Table 1 each layer of the model is made of linear elastic material, but when all layers are coupled in one composite structure, the whole model responses as a nonlinear composite, just as the arterial wall does.

Based on previous experiments, in normotensive condition the thickness of L_2 layer is three times the thickness of L_1 layer [Qiu and Tarbell 2000]. Since remodeling is known to occur by stiffening in collagen content of artery, Young's

modules of Elastic lamellae (L_1) is assumed to be constant and equal to 50^{kPa} as elastin's Young's modules. Knowing the overall mechanical properties of the artery and mechanical properties for L_1, Young's modules for L_2 layers were extracted in respect to geometric portions of L_1 and L_2 layers in arterial wall.

Geometry modeling and boundary conditions

The media of the arterial wall, as the major contributor to load bearing, is modeled by application of finite element method using ANSYS software version 11. The media has been considered to include different layers with two material properties, and uniform distribution of layers across the thickness [Dobrin 1999]:

1. Elastic lamellae (L_1) which are mainly composed of elastin fibers
2. Inter lamellar zone (L_2) with combination of collagen fibers, fine elastin fibers, smooth muscle cells, and ground Substances. (Figure 1)

The model is shaped by 2D, plane 82 elements capable of large deformation. Pressure values equivalent to mean arterial pressures of aortic sites have been applied on the inner surface and the symmetric boundary condition is considered. Because of the symmetry of the artery one forth of artery section (90 degree) was modeled and symmetry boundary condition was applied in both side in way to let the artery's radius could be changed during the analysis and pressure was applied to the inner elements.

To create different geometry, the thickness of collagenous layers which was all equal at the first, altered in a way that first inner collagenous layers thicken most and the last layer (outer) thicken least. In figure 2 the horizontal line indicates the initial thickness of collagenous layers and declined line stands for the modified thickness. The amount of thickness which added to inner layer considered as a variable called t_1 and similarly t_2 for the last layer. All the middle layers thickness were supposed to change in a way to create a line beginning from inner to outer layers thickness. Then all other geometries were created by changing t_1 and t_2.

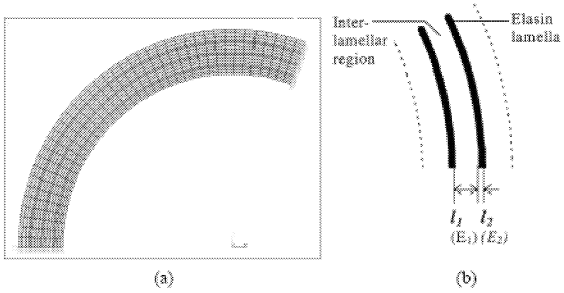


Figure 1. Finite element model using the layered concept of the artery

With the assumption of stepwise pressure to 10^{mmHg} from normotensive to hypertensive there would be 6 step and for each step more than 30 geometry was configured therefore more than 180 different geometry should modeled and analyzed. In order to avoid to repeat the time consuming procedure for modeling and applying the boundary condition of each model, a MATLAB code was programmed to link with Ansys software and the entire procedure of modeling, applying boundary condition, applying loads and running the analysis automatically performed. To prevent any convergence problems in the solution, the blood pressure was applied gradually to the model.

Table 1. Mechanical and geometrical properties of human's Abdominal Aorta [5]

PROPERTIES	QTY.
Number of all layers	57
Number of L_1 layers	28
Number of L_2 layers	29
Inside radius (mm)	8
Thickness (mm)	1.6
Young's modulus of L_1 layers (kPa)	50
Young's modulus of L_2 layers (kPa)	900

Remodeling algorithm in FE model

To study remodeling of the arterial wall in response to hypertension, the first step includes application of a normal mean arterial pressure (100^{mmHg}) to the model and calculation of circumferential stress across the arterial wall. The remodeling algorithm is such the way that by increased pressure the arterial thickness increases to compensate the elevated stress across the arterial thickness. The thickening of the aortic wall is based on the addition of collagen fibers to the interlamellar zone [Sho et al. 2001]. It has been suggested that thickening of the arterial wall due to aging and remodeling does not include addition of elastin fibers [Sho et al. 2001]. Such procedure in a lamella model would be equal to thickening of the inter-lamellae layers, while the thickness of lamellae is kept constant. By thickening of inter-lamellae layers in the composite structure of the artery, the fractional volume of interlamellar and consequently collagen increases, and hence the whole arterial wall becomes stiffer. While previous studies did not consider age related stiffening of the arterial remodeling, application of lamellar structure results in a natural algorithm of stiffening based on geometrical alterations.

The change in layer's thickness are managed in a way at which it covered all the possible configuration of modified layers in remodeling process so as to be sure of existence of an answer. Figure 2 illustrates one of the configurations in which the inner layers are more thickened than outers ($t_1 > t_2$).

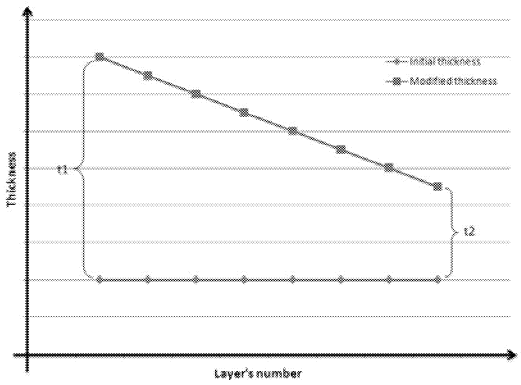


Figure 2. Thickness of the collagen layers

As simplification, the layer's thickness is supposed to change linearly through first and last layers so by dedicating these two layer's thickness the new geometry can be determined. In this paper the amount of changes of the first and last layer's thickness are considered as t_1 and t_2. It's obvious that each pair of (t_1, t_2) create a different geometry. Setup different models t_1 and t_2 was changed in a range from 0 to $45^{\mu m}$ and for each ordered pair of (t_1, t_2) the corresponding Ansys code was generated by means of Matlab software. The two softwares were link together and all generated codes transferred to Ansys program and analyzed. The post-processor was to calculate and store the new values of circumferential stress for each layer. These values are individually compared to stress values of the same layer under normotensive pressure. Then a numerical code summed up the square root of stress difference for all layer and the result stored as R variable. So the lower of R, the fitter the hypertensive stress profile to normotensive profile. Mathematically it can be said that variable R is being minimized in the remodeling process to compensate the excessive stress which induced by hypertension pressure.
For each step a database of results was created for optimization process. The optimized result of R and corresponding ordered pair of (t_1, t_2). In order to study the process of remodeling in artery the following steps are taken:

Step 1. Normal mean arterial pressure (100^{mmHg}) is applied to the lamellar model and the circumferential stress across the arterial wall is calculated. This stress is assumed as a reference stress and all stresses that will be computed are compared with the results of this analysis.

Step 2. In hypertension, the mean arterial pressure is assumed to rise to 160^{mmHg}. Thus, the luminal pressure is elevated stepwise from 100^{mmHg} to 160^{mmHg}. So the analysis is divided to 6 distinct sub steps.

Sub step 1- different possible values for t1 and t2 are generated and corresponding geometries are created.

Sub step 2- FEM analysis for different geometries related to each set of t1 and t2 is done so that corresponding circumferential stresses are computed.

Sub step 3- According to results obtained in sub step 2 and Step 1, "R" is computed. "R" is the summation of square root of stress difference for each layer between revised and initial geometry.

Sub step 4- Regarding to results obtained from last sub steps, an Artificial Neural Network System has been developed. Task of this ANN system is a function which maps thickness sets to "R" values.

Sub step 5- the function developed in the last Sub step, is introduced to Genetic Algorithm. GA seeks for values of t_1 and t_2 which minimizes the value of R. This value is stored as an answer for this pressure step.

This is important to remember that the training sets by which ANN is trained is discrete sets and the optimum point not necessarily lays in data set created in Sub step 2. So a function should be created using the input data sets so that the optimization becomes feasible.

Step 4- Above mentioned sub steps are done for each pressure increment and the optimum values are stored.

Step 5- 6 sets of t1 and t2 with their corresponding "R" value is created. A line could be fitted on this data set and a function could be computed. This function maps input pressure to optimum t_1 and t_2 value.

Artificial Neural Network (ANN)

In order to model the mathematical behavior of R (as output) subjected to the change of (t_1, t_2) (as input) in each pressure step, a Neural Network was used. After trying different network architectures and different transfer functions in MATLAB software, a 4 layer feed forward supervised network using Delta-Back propagation training rule was used. The network included an input layer with 2 neurons, 2 hidden layers, and a single output layer. All the transfer functions were Sigmoid, except the output layer with a Purelin transfer function. The network also was trained using TrainLM training function. The number of neurons in the hidden layers were different depend on the pressure step Network architecture for each pressure step in listed in Table 2.

Table 2. Number of neurons in each layer and each pressure step

	Network layers			
Pressure step	1 (Input)	2	3	4 (Output)
110	2	4	4	1
120	2	4	3	1
130	2	4	3	1
140	2	4	3	1
150	2	5	5	1
160	2	5	5	1

A total number of 36 different data sets were used to train the network, and about 8 to 12 data set as validation data. The goal of each network was set to 0.001.

Genetic Algorithm (GA)

Generally GA uses a function for optimization and by default the process directed to minimize the function. Here function F which created by NN was used by GA for optimization. As the neural networks described above are actually complicated functions indicating the relationship between (t_1, t_2) and R and cannot be minimized manually, GA was used. To obtain this, Single Objective Genetic Algorithm (SOGA) was used to find the minimum of R^2 and corresponding (t_1, t_2), in each pressure step, which are the same (t_1, t_2) of the nearest to zero R. Using MATLAB GA Toolbox, Real-Value GA was used with an initial population size of 20, Rank based fitness scaling and Stochastic uniform selection rule. An elite count of 2 and mutation and cross-over operators were used. Using the stopping criteria of maximum generations of 100, Stall generation of 50 and function tolerance set to 10^{-6}, the algorithm was run for 5 times and the global minimum of each R^2 and corresponding t_1, t_2) were found. R^2 was used to avoid negative numbers to appear in the process.

Results

Figure 3 shows the value of circumferential stress across the wall thickness in the layers of the abdominal aorta in normotensive conditions. It can be seen that the stress values in collagenous layers are markedly higher than those in the elastic layers. This is due to the higher stiffness of collagenous layers. Stress decreases monotonically from inside to the outside of the artery. Results show that the amount of excessive circumferential stress is much higher in inner collagenous layers than outer layers.

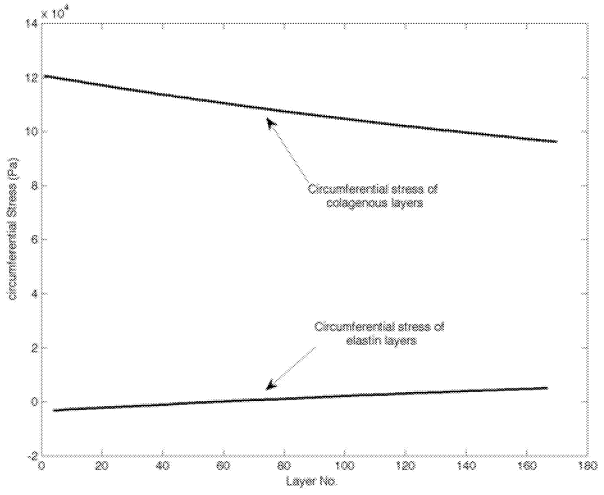


Figure 3- Circumferential stress profile of normotensive. The top line indicates collagenous layers which according to higher Young's modules should stand in the top of elastin layers (the bottom line).

It was known that the thickness of layers change during the remodeling process to compensate the effect of hypertension pressure. The remodeling process was supposed to 1: decrease the circumferential stress of layers and 2: work in an optimum way to decrease the induced pressure.
Figure 4 and 5 shows the result of fitting function (function F) of the third and sixth step (130 and 160 mmHg) which created by means of neural network. The vertical axis related to R where horizontal axis shows t_1 and t_2.

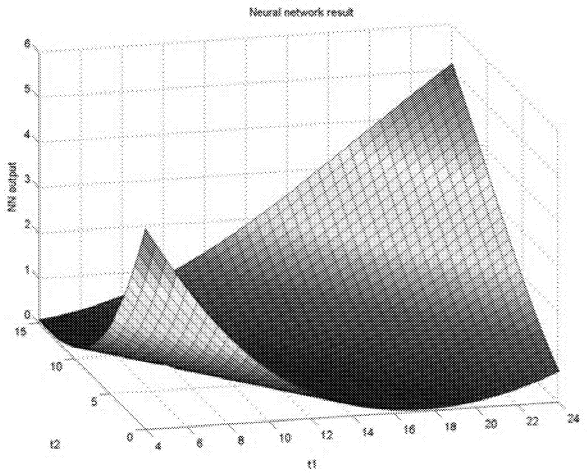


Figure 4. Result of NN for the third step (pressure=130^{mmHg})
The vertical axis which related to R, colored from blue to red (lowest to highest amount of R) for enhance visualization.

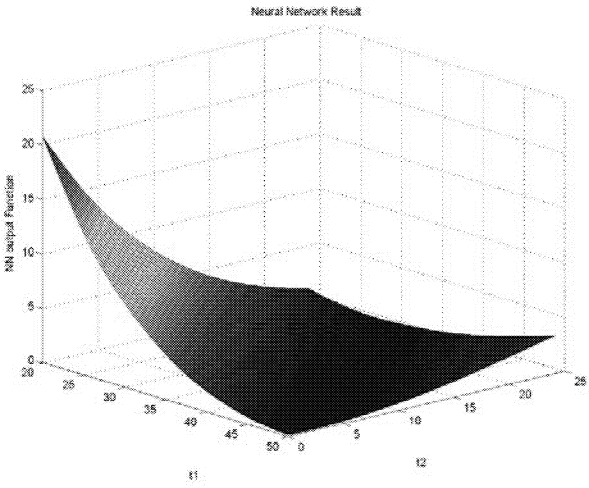


Figure 5. Result of NN For sixth step (Pressure = 160^{mmHg})

Table 3 shows the variables t_1 and t_2 in which minimum of functions occurred. In other word for each pressure step, the geometry related to ordered pair of (t_1, t_2) given in table 3 has the fittest circumferential stress answer to normotensive.

Table 3 – the result of GA
At each step, minimum of fitting function and the corresponding variables t1 and t2 are calculated.

Pressure step	t_1 (μm)	t_2 (μm)	Minimum of fitness function (R)
110	4.17	4.99	0.08565
120	16.09	1.32	0.03722
130	24.00	1.97	0.06621
140	33.94	0.67	0.07204
150	43.30	0.00	0.09665
160	50.00	1.77	0.11790

Figure 6 shows the amount of t_1 and t_2 variables obtained from optimization. As it can be seen for all of the situations except step 110, variables t_1 are higher than t_2. It also shows that t_1 and t_2 variables versus pressure in which data of variable t_1 is well fitted to ramp line and shows that variable t1 and excessive hypertension pressure are linearly related with each other

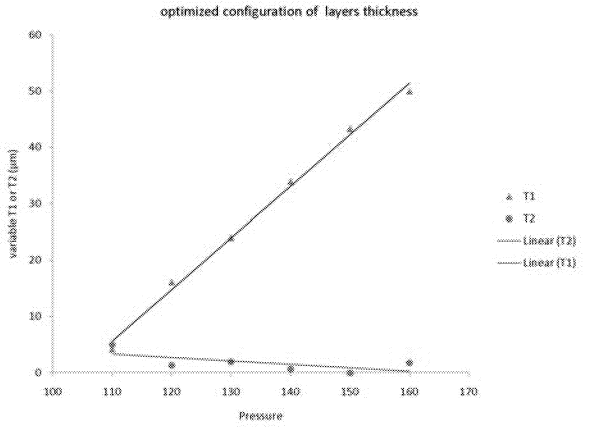


Figure 6. For each pressure step the optimum t_1 and t_2 is obtained from GA
And Points data of t_1 is seemed to be liner-like

Conclusion

In this paper, a lamellar model of hypertension abdominal aorta was considered of which the remodeling configuration to be found. Blood pressure was increased stepwise from normotensive condition. Different geometries were produces for each pressure step in ANSYS software be means of a code written in MATLAB software. The change of lamellar model layer thicknesses were used as input training data of a Neural Network to surmise the summation of difference between wall circumferential stress profile of each configuration and normotensive stress profile. The output training data also elicited from the same FEM models. The resulting function of the Neural Network was minimized using a GA algorithm in order to find the thicknesses producing the minimum circumferential stress difference between normotensive and hypertensive modes in each step. The results showed that inner layers were more thickened to compensate the excessive induced pressure, as it requires the minimum extra material to remodel the artery wall. The result also showed that the optimum thicknesses versus the pressure steps were changed nearly linear.

REFERENCES

Han HC, Ku DN, Contractile responses in arteries subjected to hypertensive pressure in seven-day organ culture, *Ann Biomed Eng* **29**(6):467–475, 2001.

Jacobsen JCB, Gustafsson F, Holstein-Rathlou NH, A model of physical factors in the structural adaptation of microvascular networks in normotension and hypertension, *Physiol Meas 24:891–912*, 2003.

Sho E, Nanjo H, Sho M, Kobayashi M, Komatsu M, Kawamura K, Xu C, Zarins CK, Masuda H, Arterial enlargement, tortuosity, and intimal thickening in response to sequential exposure to high and low wall shear stress, *J Vasc Surg 39:601–612*, 2004.

Ghiadoni L, Taddei S, Virdis A, Sudano I, Di Legge V, Meola M, Di Venanzio L, Salvetti A, Endothelial function and common carotid artery wall thickening in patients with essential hypertension, *Hypertension 32:25–32*, 1998. (14jmmb)

Taber, L.A. (1998). "A model for aortic growth based on fluid shear and fibre stresses. *J. Biomech Eng. 120*: 348- 354. (4 this paper)

Pries AR, Reglin B, Secomb TW. Remodeling of blood vessels: responses of diameter and wall thickness to hemodynamic and metabolic stimuli. *Hypertension.* 2005; 46: 725–731

Pierre Boutouyrie, MD, PhD; Caroline Bussy, MD; Patrick Lacolley, MD, PhD; Xavier Girerd, MD; Brigitte Laloux; Stéphane Laurent, MD, PhD, Association Between Local Pulse Pressure, Mean Blood Pressure, and Large-Artery Remodeling, *Circulation.* 1999;100:1387-1393

Michael R. Ward, MBBS, PhD; Gerard Pasterkamp, MD, PhD; Alan C. Yeung, MD; Cornelius Borst, Arterial Remodeling: Mechanisms and Clinical Implications, *Circulation. 2000*;102:1186

Masuda H, Zhuang Y, Singh TM, Kawamura K, Urakami M, Zarins CK, Glagov S, Adaptive remodeling of internal elastic lamina and endothelial lining during flowinduced arterial enlargement, *Arterioscler Thromb Vasc Biol 19*:2298–2307, 1999.

Qiu Y, Tarbell JM, Numerical simulation of pulsatile flow in a compliant curved tube model of a coronary artery, *J Biomech Eng 122*:77–85, 2000.

Sho E, Sho M, Singh TM, Xu C, Zarins CK, Masuda H, Blood flow decrease induces apoptosis of endothelial cells in previously dilated arteries resulting from chronic high blood flow, *Arterioscler Thromb Vasc Biol 21*:1139–1145, 2001.

Hashimoto M, Eto M, Akishita M, Kozaki K, Ako J, Iijima K, Kim S, Toba K, Yoshizumi M, Ouchi Y, Correlation between flow-mediated vasodilatation of the brachial artery and intima-media thickness in the carotid artery in men, *Arterioscler Thromb Vasc Biol 19*:2795–2800, 1999.

Nichols WW, O'Rourke MF, *McDonald's Blood Flow in Arteries*, 4th ed., Oxford University Press, New York, 1998.

Taber LA, A model for aortic growth based on fluid shear and fiber stresses, *J Biomech Eng 120(3)*:348–354, 1998.

Jacobsen JCB, Gustafsson F, Holstein-Rathlou NH, A model of physical factors in the structural adaptation of microvascular networks in normotension and hypertension, *Physiol Meas 24*:891–912, 2003.

Davies PF, Flow mediated endothelial mechanotransduction, *Physiol Rev 75*:519–560, 1995.

Langille BL, Bendeck MP, Keeley FW, Adaptations of carotid arteries of young and mature rabbits to reduced carotid blood flow, *Am J Physiol 256(4 Pt 2)*:H931–H939, 1989.

Carretero OA, Vascular remodeling and the kallikrein-kinin system, *J Clin Invest 115(3)*:588–591, 2005.

Mulvany MJ, Baumbach GL, Aalkjaer C, et al. *Vascular remodeling Hypertension 1996*; 28(3): 505-6.

Hope D. Intengan; Ernesto L. Schiffrin, Vascular Remodeling in Hypertension: Roles of Apoptosis, Inflammation, and Fibrosis, *Hypertension. 2001*;38:581

What is the full address ?

Fung YC, Biomechanics: Mechanical Properties of Living Tissues, *Springer-Verlag*, New York, 1988.

Rachev A., Stergiopulos N., and Meister, J. (1996). "Theoritical study of dynamics of arterial wall remodeling in response to changes in blood pressure." *J. Biomech. 5*: 635- 642.

Rachev A., Stergiopulos N, Meister JJ. Theoretical study of dynamics of arterial wall remodeling in response to changes in blood pressure. *J Biomech* 29: 635–642, 1996.

Rachev A. Theoretical study of the effect of stress-dependent remodeling on arterial geometry under hypertensive conditions. *Journal of Biomechanics*, Volume 30, Issue 8, Pages 819-827

Tsamis A., and N. Stergiopulos, Arterial remodeling in response to hypertension using a constituent-based model, *Am J Physiol Heart Circ Physiol*, November 1, 2007; 293(5): H3130 - H3139

Taber L.A., Humphrey J.D. Stress-Modulated Growth, Residual Stress, and Vascular Heterogeneity. *Journal of Biomechanical Engineering*, December 2001, Volume 123, Issue 6, pp. 528-535

Liu, S.Q., and Fung, Y.C. (1989). "Relationship between hypertension, hypertrophy and opening angle of zero–stress state of arteries following aortic constriction." *J. Biomech. Engr*. 111: 325- 335.

] Matsumoto, T., and Hayashi, K. (1994). "Mechanical and dimensional adaptation of rat aorta to hypertension." *J. Biomech. Engr*. 116: 278- 283.

Wolinsky, H., Glagov, S., "Comparison of Abdominal and Thoracic aortic medial structure in mammals", *Circ. Res. , Vol. XXV*, Dec. 1969, 677- 686.

Wolinsky, H., Glagov, S., "A lamellar unit of aortic medial structure and function in mammals", *Circ Res, 1967*, 20: 99-111.

JM Clark and S Glagov , Transmural organization of the arterial media. The lamellar unit revisited, Arteriosclerosis, *Thrombosis, and Vascular Biology*. 1985;5:19-34.

P. B. Dobrin, Distribution of Lamellar Deformations : Implications for Properties of the Arterial Media, *Hypertension*, March 1, 1999; 33(3): 806 – 810.

Holzapfel, G.A., Ogden, R. W., "Biomechanics of soft tissue in cardiovascular system", *Springer Wien* New York, 2003.

Nakashima T, Tanikawa J, A study of human aortic dispensability with relation to atherosclerosis and aging, *Angiology*, 1971, 22(8): 477-790

CYCLE ACCURATE SIMULATOR GENERATION USING HARMLESS*

Rola Kassem, Mikaël Briday,
Jean-Luc Béchennec and Yvon Trinquet
IRCCyN
1, rue de la Noë BP 92101
44321 Nantes Cedex 3, France
email: firstname.name@irccyn.ec-nantes.fr

Guillaume Savaton
ESEO
4, rue Merlet de la Boulaye - BP30926
49009 Angers Cedex 01, France
email: guillaume.savaton@eseo.fr

KEYWORDS
Cycle Accurate Simulator, Hardware Architecture Description Language, Pipeline Modeling.

ABSTRACT

Simulators are commonly used in embedded system development processes for early functional validation of code and exploration of new instruction set design. Such a simulator can be either hand-written or generated automatically, based on an Hardware Architecture Description Language. Automatically generated simulators are more maintainable and are faster to develop, but they also generally suffer from low performances in simulation speed and a lack of expressivity in the description. This paper presents how a cycle accurate simulator is generated automatically from a description written using the HARMLESS language. It differs from other hardware description languages in many ways: it resolves most expressivity issues and naturally offers a flexible description by explicitly splitting the syntax (mnemonic), format (binary code) and behavior descriptions. Thus, it allows an incremental description, starting for example by the disassembler (requiring format and syntax descriptions). When the first two descriptions are validated, the behavior description is added to obtain the instruction set simulator. Finally, the micro-architecture description adds information to build a cycle accurate simulator.

Introduction

The interest of the simulation techniques for the design of software, in particular for embedded systems, does not need to be justified anymore. Simulation techniques do not oppose to other V&V techniques (Verification & Validation), particularly formal V&V techniques; they are complementary of these techniques often based on large grain models. Another interest of simulation is the possibility to design and validate software when the hardware is not yet available. In this way, both software and hardware may be designed simultaneously and time-to-market is reduced.

The simulator is the central element and its specifications depend on the simulation objectives. In our application field - real-time embedded systems - we need to simulate the execution of the binary code of an application on a fine grain model of the processor. Two simulation approaches are attractive. The Instruction Set Simulator (ISS) only takes into account the instruction behavior, independently of the time needed to execute the instruction, whereas a Cycle-Accurate Simulator (CAS) takes into account timing of the real system (it models the internal architecture). Both simulation schemes are interesting. A CAS is slow but essential for real-time system simulation while ISS has better performances. Some ISS can also be associated to a structural simulator to offer both ISS and CAS advantages. The development of an *ad hoc* simulator is a difficult task, especially the simulator validation for complex modern processors. Moreover, most of this work is not reusable for a new target architecture. So, in order to simplify this task, an hardware Architecture Description Language (ADL) may be used.

The work presented in this paper is a part of a larger project aiming to the realization of a simulator offering both ISS and CAS possibilities, and used for the design of real-time embedded systems. Some information about ISS and CAS simulator generation are quickly given in this paper and the interested reader can refer to [Kassem et al. (2008)] and [Kassem et al. (2009)] for more details. This paper focuses on the automatic generation of a Cycle-Accurate Simulator from the description of the pipeline using HARMLESS (Hardware ARchitecture Modeling Languague for Embedded Software Simulation), an Architecture Description Language (ADL). HARMLESS differentiates itself from other ADL by using four independent views for binary format, syntax, behavior and micro-architecture. This approach is more flexible and eases the description. Some internal aspects on the decoder generation and the execution approach are also shown.

The paper is organized as follows: The first section is dedicated to related work on other ADL with an emphasis on LISA. The second section explains how the pipeline is modeled internally to generate the simula-

*This work is supported by ANR (French National Research Agency) under Grant Number ADEME-05-03-C0121.

tor. The third section gives an overview of the language and focuses on the micro-architecture view. In the next section, some internal aspects of HARMLESS ADL are given. Then some results are presented and the conclusion ends the paper.

Related work

Generally, an ADL must be able to specify a wide variety of architectures. An architecture description must be easy to modify and easy to understand by a designer. A lot of works has been done on ADL that can be classified into three categories: the first one that captures the Instruction Set (i.e., nML [Fauth et al. (1995)], ISDL [Hadjiyiannis et al. (1997)]); the second one that captures the structure of the processor (i.e., MIMOLA [Bashford et al. (1994)]); and the third one that mixes the two first categories (IS and structure) ((i.e., LISA [Pees et al. (1999)], EXPRESSION [Halambi et al. (1999)]). To obtain an ISS, an ADL must allow the description of the binary format, the behavior and the syntax of the instructions. In addition, in case of a CAS, it is necessary to add the description of the micro-architecture.

ISDL [Hadjiyiannis et al. (1997)] is more flexible and its semantic is stronger than those of nML. It allows the description of a wide variety of architectures with emphasis on VLIW architectures. As nML, The IS contains behavioral and structural informations. Moreover, it allows the description of the hardware of micro-architecture, that is mixed with the behavior of the instructions.

The main goal of MIMOLA [Bashford et al. (1994)] is the hardware-software co-design, this implies a much complete modeling of the structure of the processor, and gives the advantage that the same description is used for both processor synthesis, and code generation. The IS is extracted from the structure, this task can be difficult for complex instructions. Generally, MIMOLA is considered as a very low-level language and is laborious to write and modify. In addition, simulation is slow in MIMOLA environment.

In [Pees et al. (1999)], LISA is presented as a machine description language that gives a formal description of programmable architectures, their interfaces and peripherals. In many aspects, LISA includes ideas wich are analogue to nML, and the binary format, behavior and syntax of instructions are placed in the same view. LISA allows to describe a pipeline in an abstract way (i.e. the designer does not need to give the structure of the processor) and, from this point of view, is close to HARMLESS. However, HARMLESS uses a different approach. In both languages, the designer defines the pipeline stages. with LISA, pipeline registers are explicitly defined and instructions reference these registers in their behavior. As a result a change in the pipeline implementation leads to the rewriting of the instruction behavior since pipeline registers will be changed too. For each pipeline stage, the designer defines a set of *operations* that are activated when the pipeline stage is reached in the simulation. This part of the description may be complicated. With HARMLESS, the designer assigns components (see section *Hardware components description*) to the pipeline stages to control the concurrency of components accesses. Components and instruction set descriptions are independent from the description of the pipeline and a change in the pipeline does not lead to a rewriting of the instruction set description or the component description. The HARMLESS compiler maps component accesses to the pipeline stages using devices automatically (see section *Micro-architecture view*). This way, the pipeline description is simpler to design and many pipeline descriptions may be done for the same instruction set and components description.

HARMLESS is based on 4 separate views: the *format view*, the *syntax view*, the *behavior view* and the *micro-architecture view*. A set of trees composes the 3 first views that do not have the same structure. The separation of these views enables the user to choose the best structure for every view. This approach allows to ease the description of an architecture in an incremental way. So, the description becomes more flexible, and easy to modify to re-target on another micro-architecture for example. The architectural scope is not limited, it allows the description of a wide variety of architectures. In addition, it supports multi-word instructions (variable length instructions), as well as unsigned and signed data type with any number of bits. At last, the micro-architecture view allows an easy description of a pipeline and changes in the micro-architecture view have no impact on the other views. The HARMLESS's syntax is close to C language, so very easy to learn.

Pipeline modeling

The pipeline model in Harmless is done using an automaton. The goal is to shift a significative part of the computation from runtime to compile time (simulator generation). In this section, we focus on the key concepts that are used in automaton generation. Algorithms of the automation generation can be found in [Kassem et al. (2008)] and are implemented in the *p2a* tool which is a part of HARMLESS.

Sequential pipelines are considered in this paper (i.e. there are no pipelines working in parallel nor forking pipelines). The pipeline behavior is modeled in HARMLESS using an automaton, where a state represents the pipeline state at a particular time (see figure 1).

At each clock cycle, the pipeline goes from one state to another according to hazards. They are classified into three categories [Hennessy and Patterson (2001)]:

- *Structural hazards* are the result of a lack of hardware resources;

- *Data hazards* are the result of a data dependency between instructions;
- *Control hazards* that occur when a branch is taken in the program.

Control hazard are resolved in the simulator at runtime: instructions that are in the delay slot of a branch instruction are dynamically replaced by NOP instructions, if the branch is found to be taken.
Constraints resulting from *structural and data hazards* are used to generate this automaton and modeled using *resources*.

Resources

Resources are defined as a mechanism to describe temporal constraints in the pipeline. They are used to take into account *structural hazards* and *data hazards* in the pipeline.
Two types of resources are defined, *internal* and *external* resources, that model respectively statically and dynamically constraints.

Internal resources
can be compared to "resources" in [Müller (1993)]. They model structural hazards. As the pipeline state is known (*i.e.* the instructions are defined for each stage of the pipeline), then the state of each *internal resource* is fully defined (taken or available). In that case, when the automaton is built (and then the simulator), constraints described by *internal resources* are directly resolved when the set of next states is built.
As these resources are taken into account at build time (*static* approach), *no computation overhead* is required to check for this type of constraint at runtime.
For example, each pipeline stage is modeled by an internal resource. Each instruction that enters in a stage takes the associated resource, and releases that resource when it leaves. The resulting constraint is that each pipeline stage gets at most one instruction.

External resources
represent resources that are *shared* with other hardware components such as timers or memory controllers. It is an extension of internal resources to take into account resources that must be managed *dynamically* (*i.e.* during the simulation). For instance, in the case of a memory controller, the pipeline is locked if it performs a request whereas the controller is busy. Otherwise, the pipeline stage that requests the memory access takes the resource.
One interesting property of the external resources is that it allows to check for data hazards, as dependency between registers is resolved dynamically in function of the instruction flow.

Instruction class

To reduce the automaton state space, instructions that use the same resources (internal and external) are grouped to build *instruction classes*.
The number of instruction classes is limited to $2^{R_{ext}+R_{int}}$ (R_{ext} and R_{int} are the number respectively of external and internal resources in the system), but this maximum is not reached because some resources are shared by all instructions, like pipeline stages, which leads to get a lower number of instruction classes.
The use of instruction classes is required to prevent a combinatorial explosion during the automaton generation.

Overview of the language

A HARMLESS description contains many parts :

- the instruction set (the set of operations that a processor can execute);
- the hardware components used by the instructions like memory, registers, ALU, ...;
- the micro-architecture (the pipeline);
- the peripherals like timers, input/output, ...

Description of the instruction set

HARMLESS uses 3 independent views to describe the instruction set of a processor (more details may be found in [Kassem et al. (2009)]):

- the *format view* that describes the binary format of the instructions;
- the *syntax view* allows to describe the textual format of the instructions;
- the *behavior view* that binds a behavior to each instruction signature and provides a way to describe the components which are accessed by instructions.

Each view is a set of trees where a node describes a piece of format, behavior or syntax (i.e. the kind of the node). A node description conforms to the following syntax.

```
1 <kind> <name> [#<tag>] <kind_options> {
2     <description>
3 }
```

where <*kind*> can be `format`, `syntax` or `behavior`. As in a grammar specification language, HARMLESS allows to describe, for each view, whether a non-terminal node is built by aggregating sub-nodes or by selecting one node, or an aggregate of nodes, among several. By default, a non terminal node aggregates the sub-nodes. The `select` construct allows to choose one sub-node

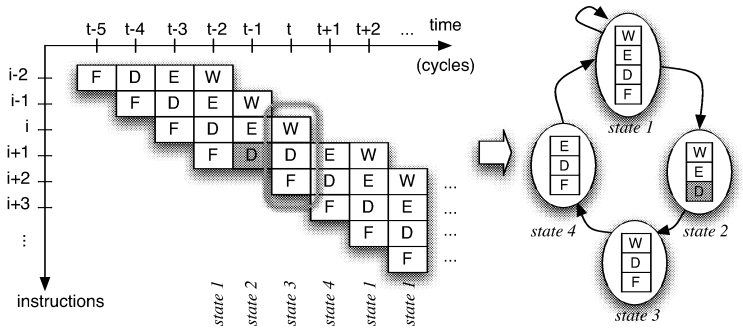


Figure 1: A state of the automaton represents the state of the pipeline at a given time. In this example with a 4-stages pipeline, three instructions are in the pipeline at time *t*, and the 'D' stage was stalled at time *t-1*. The automaton highlights the pipeline sequence, assuming that there is only one instruction type (this restriction is only made for clarity reason).

among several (or an aggregate of sub-nodes among several).
Using this model, in each view, an instruction is represented by a branch in a tree. Instructions sharing a common part in a view share nodes in the roots of the tree, while specific parts are located in leaves. A node may have a <*tag*> field. A set of tags along a branch of a tree is the unique identifier of an instruction and is called the signature of the instruction. Tags may also appear in the <*description*> part of a node and indicate a leaf in the tree.
HARMLESS is a strongly typed language. It offers signed and unsigned data type with any number of bits. As a result, the description of the instruction set is more flexible and the designer may choose the best tree structure for each view.

Hardware components description

A component is a building block of the processor architecture like registers file, memory or ALU. In HARMLESS ADL, a `component` allows the functional description of a hardware component, this description is made in an object oriented way and contains variables as well as methods. A `method` allows an instruction to access a function offered by a component.
Let's consider for exemple, a component named `Fetcher` that is in charge of the management of program counter, the description in HARMLESS is as follow (taken from the description of The Freescale XGate co-processor):

```
1  component Fetcher {
2    program counter u16 pc; -- generate get
         and set methods.
3
4    void reset() {
5      pc := 0;
6    }
7
8    void branch(s16 offset) {
9      pc := (u16)((s16)(pc) + offset);
10   }
11 }
```

This component contains a variable member which is a register (here, a specific register called program counter `pc`), as well as 2 associated methods (`reset` and `branch`).
Furthermore, HARMLESS supports the description of a complex memory mapping (for detailed informations see [Kassem et al. (2009)]). The description is currently limited to the functional part and does not include any structural information (memory hierarchy, memory latency and alignment). However, this part is easy to add and it will be included in future works.

Micro-architecture view

The micro-architecture view allows to describe the micro-architecture that implements the instruction set of the processor. The goal is to map the instruction set behavior view on the micro-architecture view using the components.
The micro-architecture is described in `architecture` and `pipeline` sections. The `architecture` section forms the interface between a set of hardware components (like registers, memory, ALU...) and the definition of the pipeline. It allows to express hardware constraints having consequences on the temporal sequence of the simulator. It may contain many `devices` to control the concurrency between instructions to access the same component.
Every `device` in the `architecture` is related to one component. The interest to differentiate the `device` from the `component` is to have the possibility of describing several devices to reach various manners to access the methods of one component.

The methods of a component can be accessed by a `port` that allows to control the competition during access to one or many methods. A port may be private to the micro-architecture or shared (i.e. the port is not exclusively used by the micro-architecture).
The `pipeline` section allows to describe a pipeline. A pipeline is mapped on an `architecture`. In the pipeline description, all the stages are listed in order. In each stage the components and the methods that can be accessed by the instruction set are enumareted. They will be mapped on devices and ports of the `architecture`.
Let's consider for instance the following micro-architecture description given by two objects (`architecture` section and `pipeline` section) using the Freescale XGate instruction set ([Freescale (2003)]).
The concurrency constraints are:

- the registers file is able to perform 3 reads and 2 writes in parallel;
- an Harvard architecture is used (separation between program and data memories);
- the computation in the ALU requires 2 stages and is not pipelined;

In HARMLESS ADL, the description of the first object (`architecture`) is as follow.

```
1  architecture Generic {
2    device gpr : GPR {
3        read  is read8  or read16
4        write is write8 or write16
5        port rs : read  (3);
6        port rd : write (2);
7      }
8    device alu : ALU {
9        port all;
10     }
11   device MEM : mem {
12       read  is read8  or read16
13       write is write8 or write16
14       shared port fetch : read;
15       shared port loadStore : read or
             write;
16     }
17   device fetcher : Fetcher {
18       port branch : branch;
19     }
20 }
```

In this description, an `architecture` named `Generic` with many devices is declared. For example, the device `MEM` controls the concurrency to access the `mem` component (*i.e.* memory) by two shared ports `fetch` and `loadStore` (*i.e.* this access can be made concurrently by other bus masters). The port `loadStore` allows to access the method `read` or `write`. The keyword `or` is equivalent to the exclusive or, and the two methods are aliases of methods declared in the `component` section (`read8 or read16` for `read` and `write8 or write16` for `write`), this aliases can be used in the `pipeline` section. This way, at a given time, if an instruction uses `read` in a stage of pipeline, the second method `write` becomes inaccessible, and the associated resource is set to busy.
Sometimes, using any method of a component makes it unavailable. Instead of forcing the user to give the list of all methods, an empty list is interpreted as a full methods list. Here the `alu` device uses this scheme.
Using HARMLESS language, the second object `pipeline` can be described as illustrated in the example below.

```
1  pipeline sample_pipe maps to Generic {
2    stage Fetch {
3        mem : read;
4      }
5    stage Decode {
6        Fetcher : branch;
7        GPR : read;
8      }
9    stage Execute1 {
10       ALU release in <Execute2> : *; --'*'
             means that any method of ALU can
             be used
11     }
12   stage Execute2 {
13     }
14   stage Memory {
15       mem : read, write;
16       GPR : read;
17     }
18   stage Register {
19       GPR : write;
20     }
21 }
```

In this example a pipeline, named `sample_pipe`, with 6 stages is declared and mapped on the `Generic` architecture. In this object, the components and their methods as well as the keywords (`read` and `write`) are used and will be mapped on the devices and ports of the `architecture` object during compile-time (see figure 2).
The 6 stages of pipeline are listed:

- a `Fetch` stage to get the instruction from memory;
- a `Decode` stage which decodes the instruction, reads operands and performs branch instructions;
- 2 `Execute` stages;
- a `Memory` access stage;
- a `Register` stage that performs write accesses on the register bank.

When using a port in a pipeline stage, it is implicitly taken at the start of the stage and released at the end. If a port needs to be held for more than one stage, the

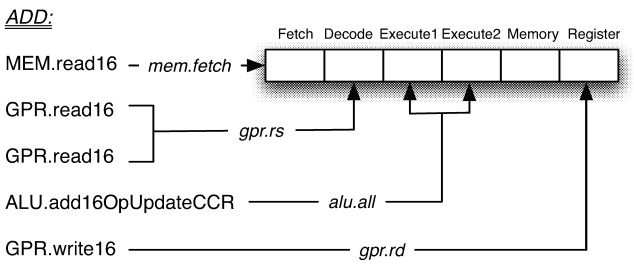


Figure 2: Mapping of the instruction set *behavior view* on the *micro-architecture view* using components access. On the left, methods of components acessed by the `ADD` instruction are shown. They are mapped on a 6-stage pipeline, using devices and ports that controls the concurrency between component accesses.

stage where it is released is explicitly given. Here port `all` of `alu` is taken in the `Execute1` stage and released in the `Execute2` stage.

Simulator generation

This section gives some details about how the simulator is automatically generated from the description. First, it introduces the model used for instructions. Then, how the micro-architecture description allows to generate the pipeline specification file (needed by the *p2a* tool) is explained, this file contains all the informations about the pipeline as the pipeline's name, the stages names, the instruction classes and so on. Moreover, the generation of data dependency controller is presented.

Instruction Modeling

The simulator is generated in the C++ language. A C++ class models each instruction. in the HARMLESS description, an instruction corresponds to a path in the corresponding tree. The C++ class that represents an instruction offers 2 main methods:

The constructor
It is associated to the decode operation. Its goal is to identify the various fields of the instruction binary code (register index, immediate, address), and stores values in the new object instance. It's important to notice that the simulator context is never affected by this operation. This function is directly linked to the format description in HARMLESS.

The execution function
It is in charge to simulate the instruction. This function is directly linked to the behavior description in HARMLESS. Even, if the behavior description is split in multiple parts to take advantages of common behaviors, this function concatenates each part of the instruction behavior description, thus removing time consuming function calls in the generated simulator. Notice that, the instructions are executed independently of the progress of the pipeline (i.e. when an instruction is fetched, it is executed directly and after mapped on the pipeline stages).

Instruction classes generation

Based on the `architecture` section and `behavior` view, instructions are grouped in classes. An instruction class puts together the instructions that use the same resources (internal and external, see section *Resources*). In our case, the external resources correspond to the shared ports and the other ports form the internal resources.
For example, using the previous description of the micro-architecture, shared ports `loadStore` and `fetch` are translated to external resources. In addition, 2 external resources are used to check for data dependancies during the simulation (in this description, an instruction can read a register in 2 stages). Other ports may or may not have an associated internal resource. For instance, the `gpr` device offers enough ports to satisfy the needs of the instruction set. So no internal resource is used to constrain the accesses to GPR's methods.
As conclusion, 11 resources are used:

- 6 internal resources for the pipeline stages;
- 1 internal resource for the ALU management: `alu`;
- 2 external resources for the memory accesses: `fetch` and `loadStore`;
- 2 external resource to check data dependancies: `dataDep`.

As each instruction depends on the `fetch` port and the pipeline stages, and the `loadStore` port is not used more than one time during one clock cycle (the access to memory can be only done in one stage), only 3 resources can differentiate instructions: there may be a maximum of $2^3 = 8$ instruction classes. From this description, we obtain 5 instruction classes. The 3 remaining instruction classes correspond to impossible configurations.

Data dependency Instruction classes generation

In the same way, based on the `behavior` view, the `pipeline` and `architecture` sections, instructions that use the same regiser ports (read or write a register) in the same stage of pipeline are grouped in data dependency instruction classes.
Using the previous description of the micro-architecture, 5 data dependency instruction classes are obtained:

- a class that groups instructions that read a register in `Decode` stage;
- a class that groups instructions that read a register in `Decode` stage and write in a register in `Register` stage;
- a class that groups instructions that write in a register in `Register` stage;
- a class that groups instructions that read a register in `Decode` and `Memory` stages;
- a class that groups instructions that read a register in `Decode` and `Memory` stages and write in a register in `Register` stage.

Data dependency controller

During the simulation, to control the access to the registers, the data dependency controller is used and works as presented in algorithm 1. An external resource is associated to a *data dependency controller*. A data dependency occurs when an instruction needs to read a register that will be updated later in the pipeline, so this instruction stalls in the current stage until it has the access right to needed data register. let's consider the pipeline described previously and 2 instructions: *ADD R1,R2,R3* in `Fetch` stage and *SUB R2,R5,R6* in `Decode` stage. The *ADD* instruction needs *R2* and *R3* to enter in `Decode` stage, but the register *R2* will be updated by the previous *SUB* instruction in `Register` stage so the request to the controller fails and the associated external resource is set to `busy`. The *ADD* instruction will be blocked in the `Fetch` stage until the *SUB* instruction writes the result in the `Register` stage.

Results

This section shows some results of the simulator build process and the computation time required to simulate a basic scenario. Special interest is ported to compare an ISS (with no temporal information) and different CAS that model 3 variations of a pipeline. The instruction set of the model is based on the Freescale XGate co-processor (16 bits RISC co-processor in the Star12X). Results are available on table 1. In the *CAS5* column, we model a basic 5-stage pipeline (`Fetch`, `Decode`, `Execute`, `Memory` and `Write Back` stages) (DLX pipeline, see Hennessy and Patterson (2001)). In column *CAS6* and *CAS7*, we split the `Execute` stage in respectively 2 and 3 stages, making a 6-stage and a 7-stage pipeline. This is done with modifying a few lines in the `pipeline` section of the description, which shows the interest of splitting the instruction set description from the micro-architecture. We give an overview of simulator performances on a simple example that calculates a Fibonacci sequence.

Algorithm 1: Instructions and data dependency controller interaction during simulation.

if *there is an instruction in the Fetch or Execute2 stage* ***and*** *the instruction will need operands in the Decode or Memory stages* **then**
- The instruction sends a request to the controller;
- **if** *at least one register is busy* **then**
  - the request fails: the external resource associated is set to `busy`;
- **else**
  - the request successes: the external resource associated is set to `available`;

else
- the request successes: the external resource associated is set to `available`;

if *there is an instruction in the Decode stage* ***and*** *the instruction will write a result in the Register stage* **then**
- The instruction sends a request to the controller;
- The controller blocks the register that will be updated;

if *there is an instruction in the Register stage* ***and*** *the instruction writes a result* **then**
- The instruction sends a request to the controller;
- The controller unblocks the updated register;

	ISS	**CAS5**	**CAS6**	**CAS7**
Description length (lines)	1267			
Time to generate ISS sources	0.43s			
Time to generate sources of pipeline model	-	0.6s	0.23s	0.89s
Number of states of the automaton	-	628	2341	8734
Number of transitions of the automaton	-	2614	10031	38274
Simulator source size (C++ lines)	13600	16000	18500	27500
Compilation time	12.1s	21.9s	22.1s	22.8s
Execution of 100 millions of instructions	*4.80s*	*25.6s*	*27.6s*	*29.2s*
Throughput (cycle/inst)	-	*1.84*	*2.15*	*2.46*
Execution of 100 millions of cycles	-	*13.72s*	*13.42s*	*12.15s*

Table 1: This table presents some results on the build process and the computation time of a simulator. `ISS` stands for Instruction Set Simulator (no timings) and `CAS` stands for Cycle Accurate Simulator. Timings are made on an Intel Core 2 Duo @ 2.4 GHz.

We can see that the simulator build process is quite fast: the simulator binary is built from the model in less than 25 s in all cases, even for a 7-stage pipeline

which requires a big automaton. For very long pipeline models, we will have to split the pipeline model in two parts, to generate 2 automata.

The most important point concerns the simulation speed, indeed, the build process is performed only once. It refers to the last three lines of the table. The computation time required for the three CAS models are in the same order of magnitude. For a given number of instructions, the model with the shortest pipeline is the faster, but it also requires less cycles for the same job. We can notice that 25.6 s are required to simulate 100 millions instructions (184 millions of cycles). As a comparison, the ISS required 4.8 s for the same scenario, but without any temporal information. So the increase factor for computing timing properties is between 5.3 and 6.1, depending of the pipeline depth.

Another point of interest is that the execution of 100 millions of cycles is a little faster on the longest pipeline. This result can be explained by the fact that there is less instruction throughput on that model (data hazard penalty increases with pipeline depth). We can notice that the computation overhead induced by the addition of the pipeline stages is extremely low, due to the internal automaton approach.

Conclusion and ongoing work

This paper has described HARMLESS ADL. One of the goal of HARMLESS is to ease the description of an hardware architecture by providing separate views for the instruction set and the micro-architecture: binary format, assembly language instruction syntax, instruction behavior and micro-architecture. Splitting the description in views allows writing it in an incremental way and lowering the amount of work that is required to get a working description.

This paper specially focuses on how the micro-architecture view is organized, and how the high level model of the pipeline is translated to an automaton. The goal of the automaton model of the pipeline is to shift a significative computation part to compile time (simulator generation), to keep runtime simulation requirements acceptable.

Currently, a working HARMLESS compiler exists and generates both an instruction set simulator and a cycle accurate simulator, based on a sequential pipeline. The performance of the simulator is good and compares favorably to existing simulators. A prototype of HARMLESS compiler may be downloaded from `http://harmless.rts-software.org`.

Future work will focus on the minimisation of the automaton and the use of multiple automata to model and simulate superscalar processors. How to model dynamic superscalar processors, including speculative execution is also planned.

REFERENCES

Bashford S.; Bieker U.; Harking B.; Leupers R.; Marwedel P.; Neumann A.; and Voggenauer D., 1994. *The MIMOLA Language Version 4.1.* Tech. rep., Lehrstuhl Informatik XII University of Dortmund, Dortmund.

Fauth A.; Praet J.V.; and Freericks M., 1995. *Describing instruction set processors using nML.* In *EDTC '95: Proceedings of the 1995 European conference on Design and Test.* IEEE Computer Society, Washington, DC, USA. ISBN 0-8186-7039-8, 503.

Freescale, 2003. *XGATE Block Guide.* Freescale Semiconductor, Inc.

Hadjiyiannis G.; Hanono S.; and Devadas S., 1997. *ISDL: an instruction set description language for retargetability.* In *DAC '97: Proceedings of the 34th annual conference on Design automation.* ACM, New York, NY, USA. ISBN 0-89791-920-3, 299–302. doi:http://doi.acm.org/10.1145/266021.266108.

Halambi A.; Grun P.; and al., 1999. *EXPRESSION: A language for architecture exploration through compiler/simulator retargetability.* In *European Conference on Design, Automation and Test (DATE).* URL `citeseer.ifi.unizh.ch/halambi99expression.html`.

Hennessy J.L. and Patterson D.A., 2001. *Computer Architecture A Quantitative Approach-Second Edition. Morgan Kaufmann Publishers, Inc.*

Kassem R.; Briday M.; Béchennec J.L.; Trinquet Y.; and Savaton G., 2008. *Simulator Generation Using an Automaton Based Pipeline Model for Timing Analysis. IMCSIT, International Workshop on Real Time Software (RTS'08), Wisla, Poland.*

Kassem R.; Briday M.; Béchennec J.L.; Trinquet Y.; and Savaton G., 2009. *Instruction set simulator generation using HARMLESS, a new hardware architecture description language.* In *SimuTools.* 24.

Müller T., 1993. *Employing finite automata for resource scheduling. MICRO 26: Proceedings of the 26th annual international symposium on Microarchitecture, Los Alamitos, CA, USA*, 12–20.

Pees S.; Hoffmann A.; Zivojnovic V.; and Meyr H., 1999. *LISA - machine description language for cycle-accurate models of programmable DSP architectures.* In *DAC '99: Proceedings of the 36th ACM/IEEE conference on Design automation.* ACM, New York, NY, USA. ISBN 1-58133-109-7, 933–938. doi:http://doi.acm.org/10.1145/309847.310101.

SIMULATION AND AI

Dynamization of core models using Description Logics and the ISITS framework

Tobias Lehmann and Andreas Karcher
Universität der Bundeswehr München
85540 Neubiberg, Germany
E-mail: {tobias.lehmann|andreas.karcher}@unibw.de

ABSTRACT

This article describes one approach how core models can be made more dynamic and more flexible for integration with other models, specialization or maintenance.
For this, two aspects are covered: first, how Description Logics can help to archive a semantic, not purely syntactic integration. After giving a brief introduction, general challenges of integration of systems are outlined as well as methods to cope with these. Then, advantages of the usage of ontologies, here in particular the Web Ontology Language OWL, are described briefly in a short example.
The second covered aspect is how the knowledge engineer bottleneck can be widened by using IT-systems domain experts are used to but actually enable them to create and modify ontological models. Here the framework for the Integration of Structured IT-Systems (ISITS) is used as generic integration framework to integration ontologies, databases, as well as semi-structured systems such as Mindmaps or Wikis. The benefits of ontologies of the first aspect mentioned are integral concept of the ISITS-framework that is – at meta level – based on technologies form Semantic Web community. Finally an example for the application of ISITS is given before the work is closed by an outlook on future activities of the authors.

INTRODUCTION

Standard models have proved to be a helpful means for the integration of distributed IT-Systems. This is true not only for databases, web services or other structured technologies in industrial environment but even more so in military environments. Often requirements towards standard models in military turn out to be highly volatile, permanently changing over time. This is because especially in non article five operations [1], governmental and non-governmental organizations are required to collaborate in changing combination in order to cope with new challenges such as met in crisis response operations or in operations to rebuild and strengthen weakening governments or economies [2]. Collaboration processes can be supported by IT-systems in many ways, but only, if the systems used can understand each other – use common models [3].
Since any computer model is designed for certain purposes the model can usually only be of great value if used in this very special context. If changing requirements shift context, once useful models turn out to lack certain perspectives or elements. A switch to more appropriate models during the usage of a model within cause massive consistency conflicts due to intensive dependencies between elements in different models.
The modification and development of concurrent model on the other hand requires expert knowledge though. But usually domain experts are not familiar with modelling and elements of any modelling language.
As the core of this work, the ISITS-framework is described which allows integration of systems using both, syntactic as well as semantic means for integration. For this, model integration and its typically associated problem types are described. Then distributed modelling is presented from the experts and standard-models- perspective as well as general clustering of common concepts. For this inheritance as known from object oriented languages is used at different levels of abstraction. As an example for the integration of a structured system with an ontology, the integration of a MySQL database is sketched. Implementations for the integration of unstructured systems such as Mindmaps, Wikis, or search engines exist too, but are not part of this document. Outlined details can be found in external papers or projects published by the authors.

Model Integration

The integration of two different models harbors several conceptual kinds of problems. If one for example plans to make two simulation systems interoperable, so that a change within one system is propagated into the other without corrupting, biasing, or loosing information transmitted, then from the semantic perspective, two layers are to be taken into consideration. This is structure as well as behavior of the elements exchanged or addressed. Behavioral aspects are usually harder to compare, to model and more complex in their interrelations to other concepts than structural aspects are. Since the focus of this paper is the latter aspect, behavioral integration is neglected here.
When information about an identical subject is exchanged, a complicated interpretation which includes a lot of not explicitly defined rules has to be conducted. Although intuitive for human actors, such an interpretation is often ambiguous due to misunderstandings and different contexts.
If exchanged subjects are not even meant to be identical by structure but by name only this problem (if not reoccurring often) is almost not solvable in an automated way.
The integration-approach used by the ISITS-framework presented in this paper will also not solve this problem, but can contribute in a small way to make it easier. ISITS makes use of technologies on a higher meta-level to decrease the effort necessary to integrate different systems in a publicly available framework and might therefore reduce required human action and interaction during integration.

The conflicts mentioned above are related to a representation through classes and instances connected by relations including inheritance. They shall be described here in more details, since they are the basis for any integration conflict. This kind of knowledge-representation has two meta-levels - namely "0" for objects, and "1", for classes [4]. It is most commonly and used in Unified Modeling Language UML [5], the XML Structure Definition Language XSD [6], or the Web Ontology Language OWL [7], and many other languages. The conflicts are clustered into the following types:

- Structural conflicts
  This very basic problem occurs, when two sets of classes which are connected via relations are to be mapped from one system to another [8]. Due to modelers' intent and different perspectives in most cases no direct class to class and relation to relation mapping is possible [9]. As depicted in Figure 1, three classes named A, B, C in one case and A, Bb as well as C1 and C2 (which are semantically equivalent to C) in the other case are connected via relations with names depending on the classes' names.

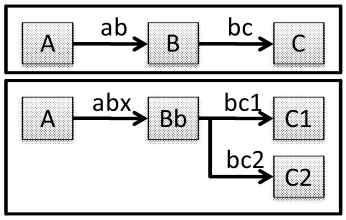


Figure 1 Structure Conflicts

  This conflict is more complicated than it might seem to be at first sight. If numbers of classes and relations increase to the thousands, then dependencies become unclear, since partial classes such as the C1/C2 case might e.g. reused in other classes and therefore are hard to compare. This conflict is addressed most directly by mechanism such as Alignment tools and Reasoners described in [10-13].

- Label-conflicts
  In general a label can be understood as a name for a thing that has a meaning [14]. This definition goes eventually back to Aristotle's semiotic triangle that connects a symbol, a thought or reference and a referent as described in [15]. If a meaning has different symbols or a symbol has different meanings (synonyms or homonyms [16]), then interpretation in different contexts mostly leads to ambiguity. This conflict appears in almost every other conflict as well, since it is a kind of basis for addressing things of a world by means of language.

- Conflicts of Integrity
  This type of conflict is probably one of the hardest ones, since it stems from implicitly defined rules and constraints which are even sometimes not part of an application's documentation. A typical representative for this is the usage of primitive data types such as floats for the formalization of particular status. If, as shown in Figure 2 a person has an attribute "mood" with range-type Float, then this is a rather common way of formalization since floats allow fuzzy calculations and are in general easy to handle (fairly high performance, directly comparable etc.). But instead of a Float type, Integer types could be used as well.

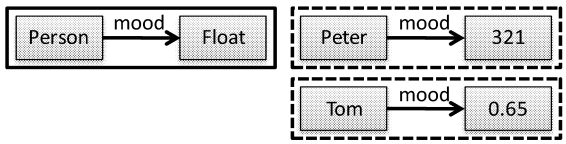


Figure 2 Integrity conflicts

  The case given above shows instances of class "Person" named *Peter* and *Tom*, both part of different systems with different integrity constraints. While Peter has mood-value 312, Tom has mood-value 0.65, which is not comparable, if the constraints are unknown. [17, 18]. Additional information such as a fuzzy function or any other means (at least documentation) of mapping numerical information to a common understanding is required.

- Conflicts at Meta structure level
  If two modeling languages are to be mapped, then the so called meta-elements of languages might differ. That is, languages allow the usage of different sets of elements such as classes, relations, instances or constraints. While e.g. OO-Languages like the .NET family or Java [19, 20] allow multiple inheritance only for interfaces and not for classes, this is an elementary feature of Knowledge Representation languages such as RDF(S) or OWL. This conflict arises, if a transformation is conducted from a language that contains more meta-elements than the target language. Therefore ISITS uses the meta-elements Web Ontology Language, a language of currently very high expressivity.

- Conflicts of data types.
  Primitive data types such as Integer numbers, Strings or Bytes are divergent over different languages. The XML-Schema Definition Language XSD for example defines Date and Time values as primitive data types while Java or .NET uses them as classes. This conflict is listed here for completeness reasons and is considered a minor problem in the context of this work.

Each of the conflicts described above requires an individual resolution. Most are addressed by technologies from the semantic web community. The projection of these technologies into existing systems that are used to model standard models or into systems experts are used to is the aim of the ISITS-framework. Before the framework is introduced, the benefits of Description Logic based modeling are outlined. As a typical example, the situation of multiple perspectives is used.

Multi-perspective-modeling

Assumed is a model for drug traffic in Afghanistan is to be created from a holistic NATO-perspective – all kinds of

things would become relevant for this model, such as the PMESII (Politics, Military, Economy, Social, Information, and Infrastructure) categories. In the example given, a model (or a set of models) would therefore have to contain information on all the six listed categories. For one category a number of Subject Matter Experts (SMEs) would be responsible to gather, formalize and maintain information. But information would surely become redudant as depicted in Figure 3: a *Warlord* is of interest for more than one category – here assumed P and M, whereas a *Drug-trafficker* is of interest for Economy and Social. In addition to this *Drug trafficker* can be *Warlords* at the same time.

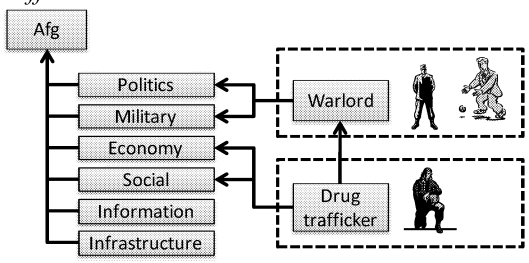


Figure 3 Redundant information within the PMESII categories

With redundant information, inconsistency will probably arise when changes in models are made without explicit knowledge of all references. A control of these in sense of defined processes are hardly enforceable in environments, where models are reused by an open community. As an alternative to redundand concepts, the Description Logics allow the definition of implicit structures, and therefore conceptually describe certain classes and general relations.

Describing implicit structures using Description Logics

Using inheritance as a means for avoiding redundancy leads – in general – to a higher reusability of models. But when different perspectives on identical classes have to be considered, the classical OO-modeling often (depending on the concrete language) lacks flexible classification and often even multiple inheritance. A next step covering this is to use the so called the Description Logics (DL). DL's purpose is to describe classes, properties and their relations in the sense of inheritance relations at structure level [21, 22]. This explicitly excludes inferences on the instance level such as provided by First Order Logic systems like Datalog, Prolog, F-Logic or comparable languages [23]. Therefore DL and its inference engines called Reasoners do not cover all needs for inference with one piece of software.
But it can be very helpful as one component for the definition of complex structures that are lateron imported by other systems such as Data Bases or OO-Languages. And this is where the benefit for the ISITS-framework is generated. The former leads to a multi-perspective on things while the latter allows standardized models to be integrated into multiple systems, all potentially benefiting from DL. In order to give a small impression on DL, an implicit PMESII-classification is described in the following:

Assumed a *Person* within Afghanistan plays an ambiguous role in the sense him to be a leader of a legal political organization as well as a *drug Trafficker* and a *Warlord*. Depending on context, this person then has to be treated differently. Therefore it is of the highest importance for each involved NATO-associated actor or NGO to know about this player's background. Within PMESII this person would be of relevance for at least four categories and therefore requires (in addition to its problematic activities) high amount of coordination between the responsible analysts. Not designating any real-world persons this particular person is labeled *Muhammad Khan* further on. *Mr Khan* is – using a DL-based KB which can be found under [24] – known to be of class *Person* only. This so called asserted relation is not even necessary, since it would be inferable from following statements assigned to *Mr. Kahn*. But for a more intuitive understandability *Mr. Kahn* has been classified as such, as depicted in the lower region of Figure 4 (each ellipse stands for a class, where each edge stands for a subclass relation).

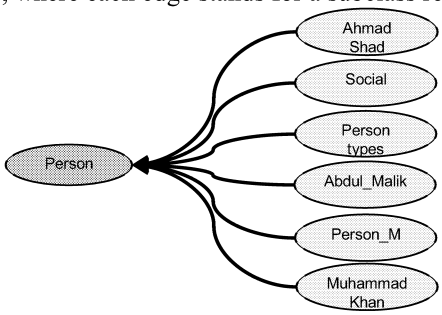


Figure 4 Asserted classification of Mr. Kahn

Each Person listed as a subclass here is described according to his/her relations. Relations in DL are defined analogously to classes, therefore can inherit from each other and connect defined domain-classes with range-classes[25]. What is known about *Mr. Khan* is that he leads *a Big Militia* (M-category) and that he conducts drug trade (E-category). To give an impression how these definitions look like, Figure 5 depicts them visualized using the Protégé-tool [26].

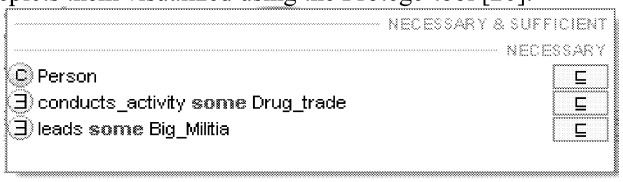


Figure 5 Description Logics Statements for Mr. Khan

The interesting point here is now, that Mr. Khan is classified according to other DL-statements stored within a different model which are too many for explanation here but again can be looked up under [24]). The result achieved in the *Person*-related section of the example model is depicted in Figure 6 and shows a fairly complicated class hierarchy containing several multiple inheritances as well as cyclic inheritance relations. On the right hand side, *Mr. Kahn* (solid rectangle) has been assigned to be a *Trader*, since he is involved in *drug trade*. Because *drug trade* is considered strongly illegal activity, *Mr. Kahn* becomes a member of the *Anti-Government* class. This makes him of interest for the P-Analyst. Each *Trader* is important for the E-Analyst. That is why he is classified as element of *Economic* importance.

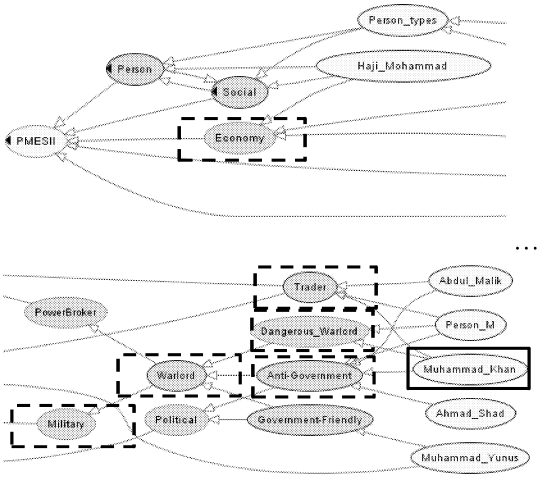


Figure 6 Inferred Classification of Afghanistan example

As a result of his role as a *Leader* of a *Big Militia* he has not only become a *Warlord* but in addition he became, founded in the potential influence of B*ig Militia*, a D*angerous Warlord*. Each *Warlord* and *Dangerous Warlord* in particular is classified as M-important. This example shows a small excerpt of the model only. What is of interest here is the fact, that relations are inferred during reasoning time and do not have to be defined statically. This means, that for example, if *Drug Trade* became changed in status from illegal to legal, *Mr. Khan* would no longer be part of the *Anti-Government* class (at least not due to his trading activities) automatically. No analyst would need to change this status from true to false manually.

The next feature DL provides, is to query a model for what class a concept really has within the context of the particular model, if only a part of its properties and restrictions are known. If a concept was at hand, from which is known only that it conducts drug trade and leads a militia, then classification-results were identical to those reasoned in the Muhammad Khan example. In return this is better means to detect similar or already existing concepts within an existing model.

It should be mentioned here, that properties play a special role in DL [27]. They can inherit from each other and therefore become important means for PMESII or any other general classification as well. A specialization example is the turnover of certain goods. Trading and production could become sub-relation of this relation, since both somewhat have to do with a certain class (e.g. Drugs are produced as well as traded). For a more detailed example see [28]. In addition to this [29] describes ways and options for a further going exploitation of DL-statements as defined by its W3C standard [25, 30]such as systems modeling.

THE FRAMEWORK FOR THE INTEGRATION OF STRUCTURED IT SYSTEMS

As presented above DL are an appropriate means for modeling and maintaining distributed KBs within heterogeneous environments. This section outlines the ISITS-fFramework which allows not only the integration of different models, based on technologies from Semantic Web Community such as publicly available standard models, or alignment tools but also a generic integration of different technological systems such as data bases, Mindmaps, XML, or ontologies. Integration in the context of this paper means propagation of changes over equal or similar structures, cloning and synchronization of structured information.

The Framework is publicly available on the code.google-site [31]. The Framework itself consists out of several abstract packages, each with one or more default implementations in order to fulfill certain functionality. As depicted in Figure 7, there are three different layers of functionality [32]. These are (from bottom to top) the Connection Layer, which defines generic access to Knowledge Bases as well as *Serializer* components and wrapper- generator to make contents available in OO-Languages.

The second layer defines allowed Elements of Communication (*EoCs*) for the interchange of structured information between different systems. EoCs are derived from meta-2 level ontology elements. This is, a reduction of all exchanged elements to classes, super class relations, instances, properties and restrictions, which define the structural level of OWL. In order to guide and coordinate messages exchanged so called Adapter are used as "wrappers" for systems. Adapters are described within the Adapter package.

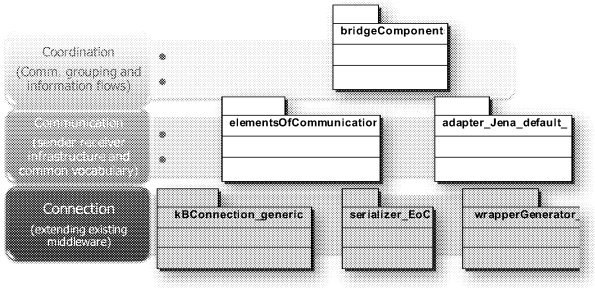


Figure 7 Framework for the Integration of Structured IT Systems

The last package to mention is the *bridgeComponent* package that allows to register and deregister adapters and to transform messages transmitted. And it allows a fast, machine aided detection of similarities between large structures based on alignment results, whilst XML-based integration can be conducted in parallel.

Without diving too deep into detail, the following brief example shows how an Ontology and a MySQL database are integrated although they do not follow identical structure definitions.

Assumed, as depicted in Figure 8, a *Request* for something, e.g. a spare part is transmitted from one system to another. The *Request* is created within an Ontology and can be transmitted to an integration *Bridge* via an *Adapter* which is called AJ in the image. AJ knows how to read within an Ontology, how to listen to change-events as well as how to propagate incoming changes into the model. By using the *EoCs*, the *Request* is broadcasted to the central *bridge*. A *Bridge* knows its registered *Adapters* and makes use of implementation of generic Transformers, which are identified by their domain and range *Adapter*-Identifiers. In this particular case, there is one implementation for the communication between AJ and AM (which adapts the MySQL database at the right hand side of the image).

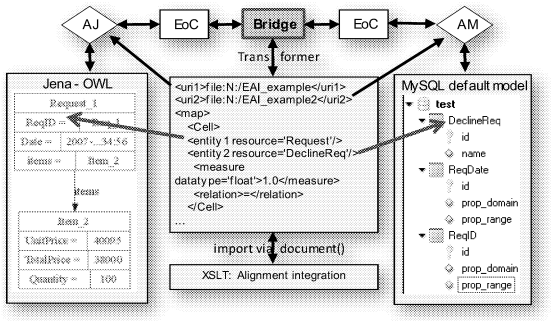


Figure 8 Model-integration using ISITS and Alignment results

There can be an unbounded number of transformers, although they are to be priorized since multiple transformations are not transitive actions. What an alignment (here used: falcon [33]) tool found out automatically is that a request-decline object is of a *superclass* of an *request* object. This very technical view leads to the possibility to create such an object just based on the existence of a request-object, which has been sent by a system unknown to the MySQL database. The mechanism behind this is not only syntactical comparison but furthermore structural comparison based on mechanisms of graph-theory as well as lexical look-ups from dictionaries such as word-net [16]. In the image's center an example-result of an alignment process is depicted, which shows an XML based map for classes and properties including a similarity measure (which is 1.0 here).

With the ISITS-framework, technically each structured or unstructured system can be brought to a level of formalization equal to one of an ontology. Unstructured models require more interpretation-effort in order to map the implicit meaning of model-elements. For example a Mindmap's elements (nodes, edges images, comments, shapes, colors etc.) can be used for any purpose. It is the freedom of a modeler to use them according to his/her intention. Because of this, there will be not only one generally applicable Adapter for Mindmaps but most likely one for each distinguishable set of usage of the Mindmap's structure-elements.

Especially in an environment, where so many similar models in different systems on different platforms are operated, this framework provides a good basis for an extensible and flexible hub and spoke integration.

OUTLOOK

The article above showed how new technologies from the Semantic Web Community can provide help within the development, maintenance and integration of standard models. Flexible structures, implemented in Description Logics, can reduce effort for integration and therefore increase reusability by enabling multiple perspectives and semantic queries through DL-statements resolved by Reasoners.

In order to bring this varieies of systems together under on common umbrella, the ISITS-framework has been created to integrate systems on the basis of common Elements of Communications. This integration is done by Adapters, which know how to address and access a model within a given system, listen to its changes and have a connection to bridges. Bridges as the hub-element within this hub-and-spoke integration architecture can transform EoCs according to mechanisms known by now and are in general open to any other future transformation which can confirm its applicability depending on two given KBs.

As an outlook the ISITS- project will further be developed, especially by uploading standard adapters and an interactive library for user-developed solutions.

REFERENCES

[1] NATO, The North Atlantic Treaty. 1949.

[2] Smith, E.A., Effects-Based Thinking: A Handbook for Whole of Government Action. 2007: Boeing, AMSE/ WSA.

[3] Bundesministerium_der_Verteidigung, Knowledge Base Development for Effects-Based Operations - Concept of Operations (CONOPS). 2006, Bundeswehr Center for Transformation - DEU-Knowledge Superiority Cell (KSC).

[4] Object_Management_Group, MOF 2.0/XMI Mapping Specification, v2.1. 2005.

[5] Object_Management_Group, UML® Resource Page. 2007.

[6] Fallside, D.C. and P. Walmsley, XML Schema Part 0: Primer Second Edition. 2004.

[7] Antoniou, G. and F.v. Harmelen, Web Ontology Language, in Handbook on Ontologies, S. Staab and R. Struder, Editors. 2003, Springer: Berlin Heidelberg.

[8] Goh, C.H., Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems. 1997, Massachusetts: Massachusetts Institute of Technology.

[9] Wache, H., Semantische Mediation für heterogene Informationsquellen. 2003, Berlin: Akademische Verlagsgesellschaft Aka GmbH.

[10] Ehrig, M. and Y. Sure, FOAM - Framework for Ontology Alignment and Mapping; Results of the Ontology Alignment Initiative, in Proceedings of the Workshop on Integrating Ontologies, B. Ashpole, et al., Editors. 2005, CEUR-WS.org. p. 72-76.

[11] Bouquet, P., et al., Specification of a common framework for characterizing alignment, knowledgeweb deliverable d2.2.1v2. 2004, Universitat Karlsruhe: Karlsruhe.

[12] Euzénat, J., et al. Ontology Alignment with OLA. in 3rd EON Workshop, 3rd Intl. Semantic Web Conference. 2004. Hiroshima (JP).

[13] Euzenat, J. and P. Valtchev. An integrative proximity measure for ontology alignment. in 1st Intl. Workshop on Semantic Integration. 2003. Sanibel Island (FL/US).

[14] Geißlinger, H., Die Imagination der Wirklichkeit. 1992, Frankfurt am Main, New York: Campus Verlag.

[15] Ogden, C.K. and I.A. Richards, The Meaning of Meaning - A Study of the Influence of Language upon Thought and of the Science of Symbolism. 1923, New York: Harcourt.

[16] Fellbaum, C., WordNet an Electronic Lexical Database. 1998, Princeton: MIT Press.

[17] Peter, B., Semantische Datenintegration: Strategien zur Integration von Datenbanken. 2005.

[18] Schmidt, A., Architekturen und Systeme zur Integration autonomer Informationsquellen. 2004.

[19] Microsoft, C# Version 2.0 Specification. 2005.

[20] Obasanjo, D., A comparison of Microsoft's C# Programming Language to Sun Microsystems' Java Programming Language. 2001.
[21] Staab, S. and R. Studer, Handbook on Ontologies (International Handbooks on Information Systems). 2004: Springer.
[22] Baader, F., et al., The Description Logic Handbook. Vol. 1. 2003, Cambridge: Cambridge University Press.
[23] Baader, F., I. Horrocks, and U. Sattler, Description Logics, in Handbook on Ontologies, S. Stab and R. Studer, Editors. 2004, Springer: Berlin Heidelberg New York.
[24] Lehmann, T., An Description Logics View on PMESII Classification in Afghanistan 2007.
[25] McGuinness, D.L. and F.v. Harmelen, OWL Web Ontology Language Overview. 2004.
[26] University, S., Protege. 2009.
[27] Bechhofer, S., et al., OWL Web Ontology Language Reference. 2004.
[28] Lehmann, T. and A. Karcher. Logic Agents and Conceptual Interoperability through Semantic Web Technologies. in 13th Americas Conference on Information Systems (AMCIS 2007). 2007. Keystone, Co.
[29] Lehmann, T. and A. Karcher. Modeling Complex Systems using the Web Ontology Language. in 5th Industrial Simulation Conference. 2007. Delft, Belgium.
[30] Motik, B., P.F. Patel-Schneider, and I. Horrocks, OWL 1.1 Web Ontology Language Structural Specification and Functional-Style Syntax - Editor's Draft of 23 May 2007. 2007.
[31] Lehmann, T., Integration of Structured IT-Systems - Google Code Project. 2009.
[32] Lehmann, T. Framework for the Integration of Structured IT Systems. in European Semantic Web Conference Workshop for Semantic Web Enabled Software Engineering. 2007. Salzburg.
[33] Jian, N.S., et al. Falcon-AO: Aligning Ontologies with Falcon. in K-Cap 2005 Workshop on Integrating Ontologies. 2005.

ANDREAS KARCHER studied Computer Science and graduated with a diploma (equivalent to a M.S.) in 1986. After 5 years of working as a project director for an IT-consultancy in Karlsruhe, 1992 he joined the Technische Universität München and received his doctoral degree there in 1996. Since February 2004 Prof. Karcher holds a tenure track University Professorship for Application Integration at the Institute of Applied Informatics at the University of the Federal Armed Forces in Munich, Germany. His main research topics are Reference Architectures, Knowledge Based Decision Support, Semantic Integration and Middleware.

TOBIAS LEHMANN studied Information Systems and graduated in 2005 from the University of the Federal Armed Forces, Germany in Munich. He then started as a PhD Candidate with focus on Decision Support Systems, Systems Integration and technical Knowledge Representation. He received his PhD in Computer Science in 2008 and is since running a Consulting Company in the field of Changemanagement in Mechatronical Products.

CONSTRAINED HANDWRITING RECOGNITION USING MODULAR KOHONEN NEURAL NETWORKS

Umran Azziz, Jaya Adiguna and Ahmed M. Zeki[1]
Department of Computer Science, KICT
International Islamic University Malaysia
Jalan Gombak, 53100 Kuala Lumpur, Malaysia
E-mail: umranium@hotmail.com, jaydiguna@yahoo.com, amzeki@iiu.edu.my

KEYWORDS
Constrained Character Recognition, Handwritten Character Recognition, Modular Neural Networks, Kohonen Self-Organizing Map

ABSTRACT

This paper describes the design and development of a system that retrieves data from handwritten forms using constrained character recognition. The system makes use of several image processing algorithms to retrieve binarized data from specially designed forms. The form is first scanned and undergoes processes such as cropping, rotation, entry areas extraction, thresholding, character extraction, fitting bounding box, resizing, and binarization. The binarized data is then fed into a modular neural network composed of several Kohonen Self-Organizing Maps for training and testing. Test results show that the system has a success rate of 84.72% and very low training and processing time.

INTRODUCTION

Character recognition is the translation of images of text (handwritten, typewritten, or printed) to machine-editable text based on the static shape of the character itself (Cheritet et al. 2007).

As a more specific form of character recognition, Handwritten Character Recognition (HCR) deals specifically with handwritten texts. HCR systems are useful in applications that require processing large volumes of handwritten data such as: processing checks in the banking industry (Guillevic and Suen 1997), sorting mail in the postal service (Maszewski and Miciak 2008), and converting documents to the digital counterparts, to name a few.

Because of the complexity introduced by the differences in handwritings, constrained handwritten character recognition introduces some constraints in order to simplify the task. In this project, the form to be filled is designed with rows of boxes in which each character is written in a single box.

An Artificial Neural Network (ANN) is a system of interconnected simple processing units, which resemble the neurons of the brain in their capability of storing experimental knowledge and making it available for use (Haykin 1999).

Modular Artificial Neural Networks (MOD-ANN) are ANNs composed of smaller ANNs, each of which is an independent module that performs a portion of the overall task (Happel and Murre 1994). The MOD-ANNs exploit the capabilities of the principle of divide and conquer in the design and learning of the modular ANNs.

MOD-ANN can be applied in any application which uses conventional ANN, such as in general pattern recognition, but with more added benefits. This is because MOD-ANN has the ability to reduce a large, unwieldy NN to smaller, more manageable components (Azam 2000). This compartmentalization translates to a more efficient, accurate, and robust ANN, as will shown in the test results.

A Self-Organizing Map (SOM) is a NN which has neurons placed as nodes of a lattice and after training by competitive learning, becomes characterized by the formation of a topographic map of the input patterns in which the coordinates of the neurons in the lattice are indicative of the statistical features contained in the input patterns (Haykin 1999). The project described in this paper makes use of the SOM model introduced by Kohonen (1982).

SOMs are useful as statistical tools for simplifying high dimensional data to lower dimensions for viewing purposes. In the case of this project it is used to simplify the 2-D bitmaps of characters obtained from a form to a 1-D character representation of each bitmap. Among other applications of SOMs is in the prediction of oil temperature of a substation transformer (Ohkita 2004).

In a recent paper (Azziz et al. 2009) the authors evaluated four ANNs, namely: Feed Forward (FF) ANN, Kohonen Self Organizing Map (SOM), Modular FF ANN and Modular Kohonen SOM. The experimental results showed that the best one is the Modular Kohonen SOM because of its high accuracy and less training duration requirement, hence it is proposed to be used in form processing systems as the case of this poject.

Sarfraz et al. (2005) used syntactic and ANN approach to recognize Arabic characters and suggested that accuracy of ANN may be increased by improving the preprocessing stage. A more recent work proposed a new method of handwriting recognition by using a combination of ANNs

[1] Corresponding Author

and genetic algorithms which achieves a 71% success rate (Mathur et al. 2008). The networks were designed with quadruple layered ANN with 625 input and 26 output neurons each corresponding to a character from A-Z, the outputs of all the four networks is fed into the genetic algorithm which was developed using the concepts of correlation.

Ali et al. (2004) proposed a technique that recognizes handwritten texts by breaking the ligature into basic geometrical strokes that are fed into a SOM. The technique achieved 85% success rate for English alphabets. Oh and Suen (2002) compared between the conventional Feed Forward (FF) ANN and modular FF ANN using English uppercase alphabets. They achieved 81% for conventional FF-ANN and 91% for modular FF-ANN. The experimental results proved superiority of the modular ANNs in terms of recognition capability. Their results showed that the performance of ANNs can be increased by modularizing them.

This work proposes the use of a Modular NN composed of several Kohonen Self-Organizing Maps for training and testing. The rest of this paper is organized as follows: next section will describe how the system is implemented, followed by how the training and testing is conducted. The experimental results will then be presented, and the paper will end with a conclusion and future research direction.

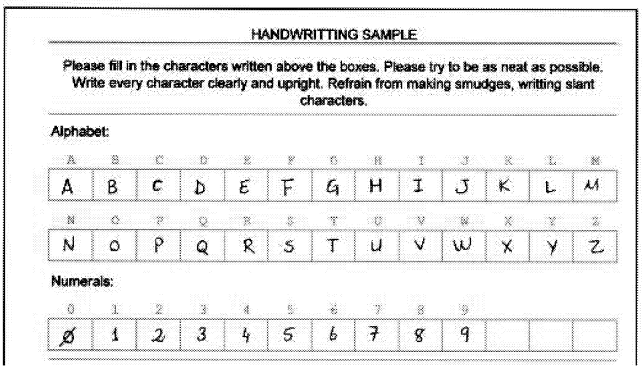
IMPLEMENTATION

After a form containing the handwritten characters is scanned, the image of the form is first applied to a number of preprocessing stages. The results are then saved to be used later for either training or testing the ANN. For this project, only characters A-Z (capital) and digits (0-9) are used. An input size of 16x16 pixels (256 values) was used for training and testing.

Design of the Form

A form was designed to collect handwriting samples of 26 uppercase English alphabets and 10 digits. The form has several requirements to facilitate the preprocessing stage. These requirements are: a plain white background, character input boxes with a light colored border, and a dark colored page border. See Figure 1.

The plain white background is to allow for more accurate results after thresholding. The character input boxes serve as a guide to write the characters. Each input box is filled with one character. Contiguous character input boxes make an entry area. The border of boxes is in light color while the characters are filled using a dark colored pen so as to easily separate the two during the thresholding stage. The dark colored page border is to allow for easy angle correction and cropping. Each form contained three handwriting sets to be filled by three different people.



Figures 1: Sample Form

Preprocessing Stages

Preprocessing stages play a crucial role in preparing the input to the recognizers for best and more accurate results. The preprocessing stages implemented in this project are: cropping & rotation, entry areas extraction, thresholding, character extraction, fitting bounding box, resizing to fix ANN input size, and binarization. The structure of these initial stages is illustrated in Figure 2.

The form image is first analyzed to check whether it is slanted or not. If the image is slanted then it is rotated to an upright position. Hough transform is used to determine the angle of rotation (Gonzalez and Woods 2002). Then, the image is cropped based on the borders of the form. Using a pre-prepared template, entry areas where characters are written are extracted. Each entry is then further processed in the next stage.

Each entry area is converted from RGB color to HSL because it describes perceived colors more accurately, while being simple (Gonzalez and Woods 2002). Only the luminosity (L) values are then taken and used as grayscale values. Otsu's method is then used to calculate the threshold level (Otsu 1979). This method assumes that the image to be thresholded contains two classes of pixels (foreground and background), then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal. At the end of this process, the resulting monochrome images are then filtered using the median filter so as to remove salt and pepper noise (Gonzalez and Woods 2002).

The entry area image is then divided into regions with characters. Each region contains only one character. The characters are then extracted from each of these regions. For each of the extracted character, the smallest bounding box that fits the character is found. This is to remove the extra whitespace surrounding the character. For each character, the smallest box is found and the character image is then resized to 16x16 pixels in order to fit the ANN input. The bicubic resampling method is used so as to maintain the fine details of the character image (Keys 1981).

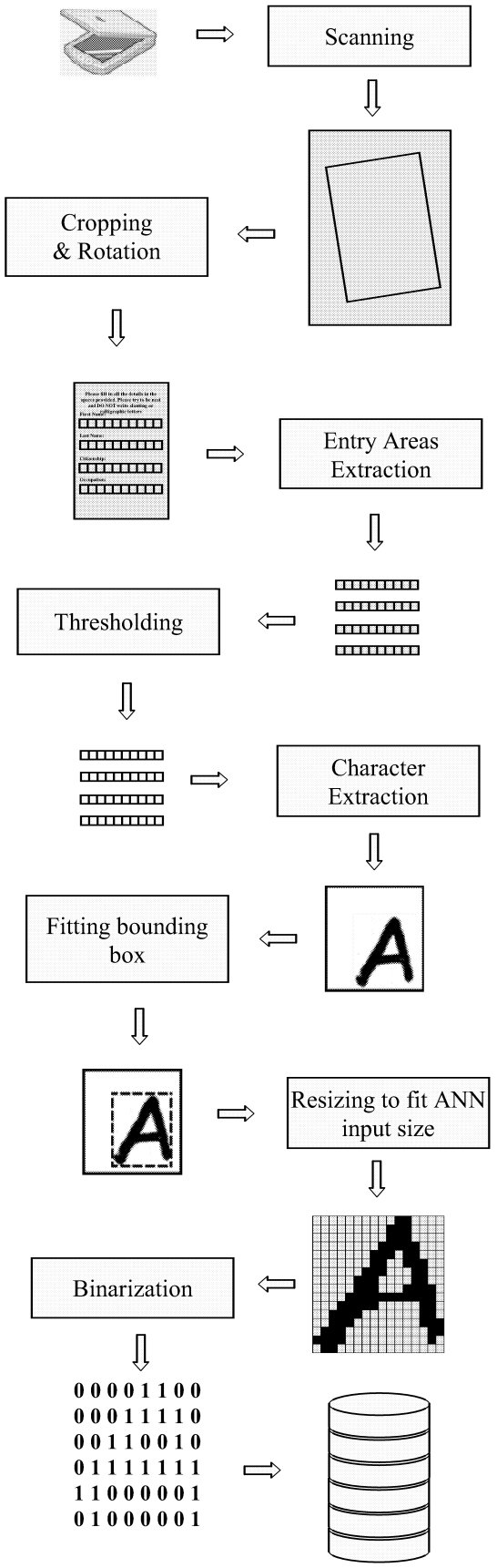


Figure 2: Scanning and Preprocessing Stages

In the binarization process, the resulting character image is then converted into an array of 0s and 1s. The array is stored for later application to the ANN for training and testing purposes.

Design of the Modular Kohonen SOM

Each component Kohonen SOM in the Modular Kohonen SOM (MOD-SOM) is trained using one handwriting set. The maximum of the results of all the ANNs is then taken as the "winner". Hence for the *N* handwritings used to train, there should be *N* Kohonen SOMs. Each Kohonen SOM has an input of 256 values and output of 36 values (A to Z, and 0 to 9). The MOD-SOM is illustrated in Figure 3. The java code for the Kohonen SOM was obtained from the book of Heaton (2008).

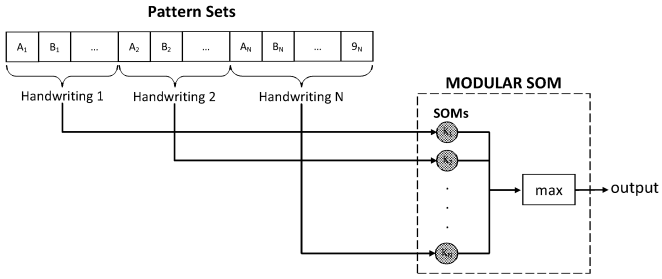


Figure 3: The Modular Kohonen SOM

Training and Testing

Convenience sampling was used to collect the data. A total of 81 handwriting sets were gathered, each containing 36 characters consisting of the uppercase English alphabet and numerals. To standardize the input data, people filling the form were requested to slash through their zeros "Ø" to differentiate it from the alphabet "O", and to add serifs to their alphabet "I" to differentiate it from the numeral one "1".

The images were then fed into the system. Approximately 75% (61 sets) of the total were used for training while the rest (20 sets, which makes 25% of the total images) were used for testing.

EXPERIMENTAL RESULTS

Each of the forms collected is first scanned using a scanner at 300 dpi. A success rate of 84.72% was obtained. Training the MOD-SOM took an extremely short time of 2.47 minutes while testing took 0.63 minutes. This is because the complexity of the problem of identifying the characters is simplified by breaking the problem down into smaller problems. By using a confusion matrix, it was discovered that the errors are due to confusion between characters with similar appearance, for example: Z and 2, J and T, G and Q, G and O, O and D, R and K. These errors occur because different people write these characters in different ways. Figure 4 shows examples of character images that were misrecognized by the ANNs.



Figure 4 - Misrecognized Character Images (2, 4, K and Z)

Figure 5 shows some characters where successfully recognized.



Figure 5: Examples of successfully recognized characters

CONCLUSION

This paper has described how a system was designed and implemented to retrieve data from handwritten forms. The system uses a NN that is designed based on modular NNs but making use of the Kohonen Self-Organizing Map to produce excellent success rates and low training durations.

Based on the results achieved, it can be deduced that MOD-ANNs simplify the complexity of training while increasing the performance of their non-modular counterparts. Using 60 handwriting sets to train and 21 to test, the system was able to achieve a high success rate of 84.72% and very short training duration of 2.47 minutes. This short training duration means that even more training can be done, hence resulting to even better success rates. A comparison with three other types of NNs was done.

For future work, further preprocessing stages such as segmentation, thinning, character normalization, and character slant correction are recommended. More training can be done using more samples hence increasing the success rates. Due to the architecture of the MOD-SOM, the training can also use computer clusters to distribute the load and hence speed up the process.

In addition, more work needs to be done to further automate the system. Such work may include interfacing the system to a scanner with an automatic document feeder and connecting the system to a database where it can save the retrieved outputs.

REFERENCES

Ali, A., Ahmad, M., Rafiq, N., Akber, J., Ahmad, U., and Akmal, S. 2004. "Language Independent Optical Character Recognition for Handwritten Text." In *Proceeding of the 8th International Multitopic Conference (INMIC'2004)*, 79-84.

Azam, F. 2000. "Biologically Inspired Modular Neural Network." Ph.D. Dissertation, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Azziz, U., Adiguna, J. and Zeki, A.M. 2009. "Performance Evaluation of Four Neural Networks for Handwriting Recognition." Submitted to *International Conference on Robotincs, Visioin, Signal Processing and Power Applications.* 19-21 Dec 2009, Langkawi, Malaysia.

Cheriet, M., Kharma, N., Liu, C., and Sun, C.Y. 2007. *Character Recognition System: A Guide for Students and Practitioners*. New Jersey: John Wiley & Sons Inc.

Gonzalez, R.C., and Woods, R.E. 2002. *Digital Image Processing*, (2nd ed.). New Jersey: Prentice Hall.

Guillevic, D. and Suen, C.Y. 1997. "HMM word recognition engine." *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, Ulm, 18-20 Aug. 1997. Vol. 2, 544-547.

Happel, B. and Murre, J. 1994. "The Design and Evolution of Modular Neural Network Architectures." *Neural Networks*. Vol. 7, No. 6-7, 985-1004.

Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation* (2nd ed.). New Jersey: Prentice Hall International, Inc.

Heaton, J. 2008. *Introduction to Neural Networks for Java*, (2nd ed.). St. Louis: Heaton Research Inc.

Keys, R.G. 1981. "Cubic Convolution Interpolation for Digital Image Processing." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 29, no. 6, 1153-1160.

Kohonen, T. 1982. "Self-organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics*, No. 43, 59-69.

Maszewski, M. and Miciak, M. 2008. "Character Recognition Based on Fourier Transfrom and CDWT in Postal Applications." In *Innovations in Hybrid Intelligent Systems. Series: Advances in Intelligent and Soft Computing*, Corchado, E., Corchado, J.M. and Abraham, A. (Eds.). Springer. Vol. 44, pp. 462-469.

Mathur, S., Aggarwal, V., Joshi, H., and Ahlwat, A. 2008. "Offline Handwriting Recognition using Genetic Algorithm." In *Advance Research in AI*, Markov, K. Ivanova, K. and Mitov, I. (Eds.). Vol. 2, No. 2, 21-27.

Ohkita, M. Du, H., Inui, M., and Kobayashi, T. 2004. Application of Self-Organizing Map (SOM) to Prediction of Oil Temperature of a Substation Transformer. *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, Vol. 104, No. 139, 89-94.

Oh, I.S. and Suen, C.Y. 2002. "A Class-modular Feedforward Neural Network for Handwriting Recognition." *Pattern Recognition*, Vol. 35, No. 1, 229-244.

Otsu, N. 1979. "A Threshold Selection Method from Gray-level Histograms." *IEEE Transactions on Systems Man and Cybernetics*, Vol. 9, No. 1, 62–66.

Sarfraz, M., Zidouri, A., and Nawaz, S. N. 2005. "On Offline Arabic Character Recognition." In Computer-Aided Intelligent Recognition Techniques and Applications. Sarfraz, M. (Eds.). Great Britain: John Wiley & Sons Ltd.

AUTHORS BIOGRAPHY

UMRAN AZZIZ was born in Mombasa, the republic of Kenya. Currently he is a final year student at the Department of Computer Science, Faculty of Information and Communication Technology, International Islamic University Malaysia. He participated in several programming contests and achieved excellent results. He also assists researchers at the department as a research assistant and lecturers as a tutor. This paper is part of his final year project.

JAYA ADIGUNA was born in Jakarta, Indonesia. He is currently a final year computer science student in the International Islamic University Malaysia (IIUM). He participated in various competitions including programming onces and won some of them, such as the ACM-ICPC Malaysia Al-Khawarizmi Programming Competition 2007 and 2008. His main interests are computer programming,

artificial intelligence, and open source technology. This paper is part of his final year project.

AHMED M. ZEKI was born in Iraq in 1972. He obtained BSc in Mathematics from the University of Jordan, Amman, Jordan, MSc and PhD in Computer Science from the National Universiy of Malaysia (UKM), Bangi, Malaysia. Currently he is an assistant professor at the Department of Computer Science, International Islamic University Malaysia (IIUM). He joint IIUM in 1999. His area of interst includes Pattern Recognition, Data Mining, and Machine Learning.

DESIGN AN APPROACH FOR REPRESENTING MAMDANI FUZZY RULES BY SUGENO RULES DEPENDING ON SINGLE PERCEPTRON NETWORKS

Rasha E. Karakchi
Department of Computer and Information Engineering
University of Mosul
Mosul
Iraq
rasha_elham@yahoo.com

KEYWORDS: Single-Neuron Perceptron, Sugeno Fuzzy, Mamdani Fuzzy, nntool.

ABSTRACT
Today, many systems designers use software tools such as Matlab to model a complex, mixed- technology system prior to physically building and testing the system. These tools, along with their associated toolboxes provide an effective means for the initial modeling and simulation stages in project. Such these toolboxes are Fuzzy toolboxes (Fuzzy tool) and Neural toolboxes (nntool) in Matlab. This paper describes an approach for representing Mamdani Fuzzy rules by Sugeno rules (linear equations) using Single Neuron Perceptron. Whereas, each fuzzy rule can be represented by four fuzzy outputs and each one of these values can be implemented as targets with its crisp inputs in Single Neuron Perceptron. Whereas, each Perceptron represents a linear equation (Sugeno equation). Using Neural toolboxes in Matlab, the values of weights and biases for each Perceptron are computed. Later, design particular architecture to implement this approach into digital logic hardware for implementation on devices such as FPGA. That architecture consists of several components with block RAMs for storing the values of computed weights and biases in addition to store fuzzy rules.

INTRODUCTION

There are many types of neural network learning rules. They fall into three broad categories: Supervised learning, unsupervised learning, reinforcement (or graded) learning (Martin T. Hagan et al. 2002).
In **supervised learning**, the learning rule is provided with a set of examples (the training set) of proper network behavior:

$$\{p_1, t_1\}, \{p_2, t_2\}, \ldots\ldots \{p_Q, t_Q\}$$

Where p_q is an input to the network and t_q is the corresponding correct (target) output. As the inputs are applied to the network, the network outputs are compared to the targets. The learning rule is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets. The Perceptron learning rule falls in this supervised learning category. **Reinforcement learning** is similar to supervised learning, except that, instead of being provided with the correct output for each network input, the algorithm is only given a grade. The grade (or score) is a measure of the network performance over some sequence of inputs. This type of learning is currently much less common than supervised learning. In **unsupervised learning,** the weights and biases are modified in response to network inputs only. There are no target outputs available. At first glance this might seem to be impractical.

PERCEPTRON ARCHITECTURE

The general Perceptron network is shown in Figure1.
Perceptron Learning Rule: If we want to examine the performance of Perceptron networks, we will introduce the Perceptron learning rule. This learning rule is an example of supervised training, in which the learning rule is provided with a set of examples of proper network behavior:

$$\{p_1, t_1\}, \{p_2, t_2\}, \ldots\ldots \{p_Q, t_Q\}$$

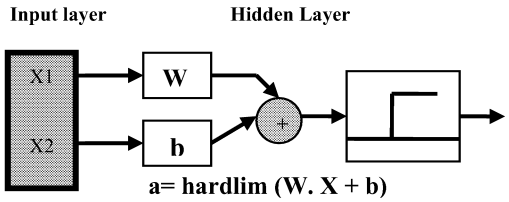


Figure 1: General Perceptron Network

$$a = \text{hardlim}(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where p_Q is an input to the network and t_Q is the corresponding target output. As each input is applied to the network, the network output is compared to the target. The learning rule then adjusts the weights and biases of the network in order to move the network output closer to the target.
Mamdani Fuzzy Control: Fuzzy controllers have found applications in many diverse fields. Fuzzy logic is an efficient way to deal with imprecise and ill defined data. Fuzzy controllers have proven to be a good tool for real time industrial processes, where it is difficult to obtain a mathematical model for the system and/or the control variables are too hard to measure (Nesar Ahmad and P. Robin Jacob, 2004). Fuzzy controllers model the human operator instead of the process itself. A defined set of rules is used for that purpose and it is called the knowledge base of the fuzzy controller. Knowledge base is a set of "IF-THEN" conditional statements (S. Singh and K. S. Rattan, 2003) .The k^{th} rule of the system can be described as

IF (a_1 is A_1) AND (a_2 is A_2). . AND (an is A_n) THEN (p is B)

Where a, and A, are defined in the universe of discourse A' while p and B, are defined in the output universe of

discourse B'. The inference engine, the other part of the fuzzy controller, is responsible for making fuzzy decisions based on the given input and the knowledge base Fuzzy encoder (Fuzzifier) and fuzzy decoder (Defuzzifier) are used to communicate with the real world Being real time systems, fuzzy controllers have to process data in short time. Hardware implementation is the natural route to high speed operations. The speed of the fuzzy controller is measured in Fuzzy Logic Inferences per Second (V. GEORGESCU, 2002). Since 1985, great deal of work has been done in hardware implementation of fuzzy controllers.

Sugeno Fuzzy Control: One type of output fuzzy sets can be Singletons, but they can also be linear combinations of the inputs, or even a function of the inputs. The general Takagi-Sugeno rule structure is

If f (e_1 is A_1, e_2 is A_2, .. e_k is A_k) then y= g(e_1, e_2,....)

Here f is a logical function that connects the sentence forming the condition, y is the output, and g is a function of the inputs. A simple example is

If error is zero and change in error is zero then y= a × error +b × (change in error) +c

Where a, b and c are all constants. This is a first order model. Inference with several rules proceeds as usual, with a firing strength associated with each rule, but each output is linearly dependent on the inputs (Jan Jantzen, 1998). The output from each rule is a moving singleton, and the defuzzified output is weighted average of the contributions from each rule. The controller interpolates between linear controllers; each controller is dominated by a rule, but there is a weighting depending on the overlap of the input membership functions. This is useful in a nonlinear control system, where each controller operates in a subspace of the operating envelope. One can say that the rules interpolate smoothly between the linear gains. Higher order models are also possible. This paper introduces a method for representing a Fuzzy Knowledge Base into linear equations (Sugeno Type) using Single Perceptron Neurons for each rule. Then, designing a particular architecture for this algorithm using FPGA and implementing it on a certain control application (Yen and R. Langari, 2003) (V. Salputra, 2000).

Algorithm of the project

The idea of this project is implemented on a certain fuzzy control application which consists of two inputs and one output as shown in figure 2. This project is implemented in several steps.

Step1: Selecting the values of x1, x2 pointed in figure 2.

X_1= 3 v, X_2= 0.5 v

Step2: As result of the overlapping between two fuzzy sets (Neg, Zero for input1, and Zero, Pos for input2). There are two degrees and labels for each input as shown in figure 2. These values are computed theoretically depending on the equation (2, 3).

d1 = (x_1- p1) × slope, d2= 1- d1 (2)

l1= 2, l2=l1-1 = 1 (depending on x1)

d3 = (x_2- p1) × slope, d4= 1- d2 (3)

l3= 1, l4=l3-1 = 0 (depending on x_2)

d1 = 7C $_h$, d2 = 83 $_h$, d3 = 3E $_h$, d4 = C1 $_h$

Step3: Finding the minimum degrees by comparing the degrees of input1 with input2's degree.

C1 = 3E $_h$, C2 = 7C $_h$, C3 = 3E $_h$, C4 = 83 $_h$

Step4: finding the output labels of the system depending on Mamdani fuzzy rules shown in Table (1).

L1 = 2, L2 = 2, L3 = 1, L4 = 2

Using equation (4), Mamdani fuzzy outputs would be obtained as shown in figure 3.

$$O_i = \frac{C_i}{\text{Slope}} + p_1 \quad (4)$$

Where i is number of fuzzy outputs, O_1, O_2, O_3, O_4 are fuzzy outputs.

O_1 = A $_h$, O_2 = C $_h$, O_3 = 2 $_h$, O_4 = C $_h$

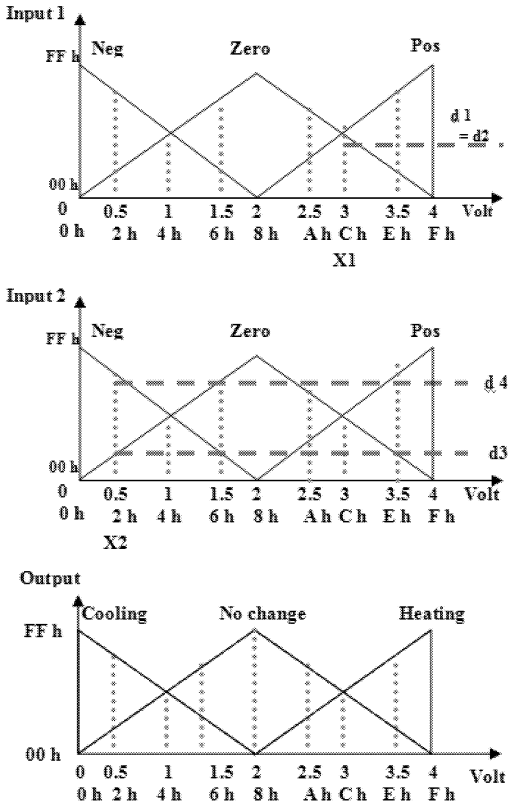


Figure 2: Fuzzy Sets of each input and output of Fuzzy Control System

Step5: Depending on Sugeno fuzzy control, this output area can be sampled into four linear equations (4) represented by fuzzy outputs as shown in figure 3 (colored lines).

$$O_1 = a_{11} x_1 + a_{12} x_2 + c_1 \quad (5)$$
$$O_2 = a_{21} x_1 + a_{22} x_2 + c_2$$
$$O_3 = a_{31} x_1 + a_{32} x_2 + c_3$$
$$O_4 = a_{41} x_1 + a_{42} x_2 + c_4$$

Table (1): Fuzzy Rules with labels of inputs and output

X1 X2	Neg 0	Zero 1	Pos 2
Neg 0	Cooling 0	Heating 2	Heating 2
Zero 1	Cooling 0	No change 1	Heating 2
Pos 2	Cooling 0	Cooling 0	Heating 2

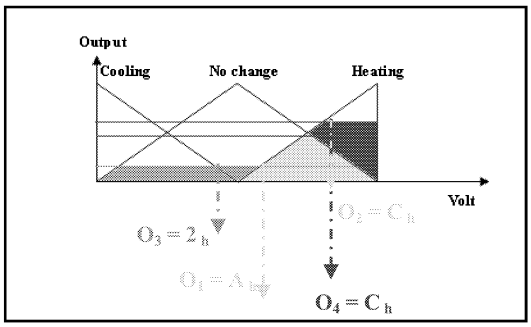


Figure 3: Fuzzy Outputs

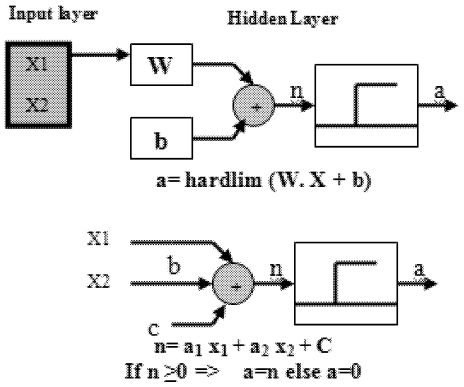


Figure 4: Sugeno Equation represented by Single Neuron Perceptron

As shown in figure 2, the range of inputs and outputs is [0, 4] volt. Each one of these two inputs is represented by 4-bit with three fuzzy sets (Neg, Zero, Pos).Table (2) shows the fuzzy outputs for each values of inputs and values of parameters trained using nntool respectively. Whereas, a_1, a_2, a_3 represent weights (w_1, w_2) and bias (b) respectively. For storing in Block RAMs, each one of these parameters is multiplied by 32 to represent as 5 bit in Block RAMs after converting the values into hexadecimal. As shown in Table (2), the weights and biases of each Neuron representing Fuzzy output are found.

$O_1 = A_h = 2.5$ v => a_1= -0.00094, a_2 = 0.96115, a_3 = 0.81761
$O_2 = C_h = 3.0$ v => a_1= 0.96397, a_2 = 0.12051, a3 = 0.047847
$O_3 = 2_h = 0.5$ v => a_1= 0.23227, a_2 = -0.00144, a_3 = -0.19605
$O_4 = C_h = 3.0$ v => a_1= 0.96397, a_2 = 0.12051, a_3 = 0.047847

Each weights and biases are multiplied by 32 for storing in Block RAM as 5-bit.
O_1 => $a_1 = 0_h$ $a_2 = 1E_h$ $a_3 = 1A_h$
O_2 => $a_1 = 1E_h$ $a_2 = 3_h$ $a_3 = 1_h$
O_3 => $a_1 = 7_h$ $a_2 = 0$ $a_3 = 1A_h$
O_4 => $a_1 = 1E_h$ $a_2 = 3_h$ $a_3 = 1_h$
To find Sugeno o outputs, equation (5) is implemented

$Y_1 = 0 \times C_h + 1E_h \times 2_h + 1A_h = 56_h$
$Y_2 = 1E_h \times C_h + 3_h \times 2_h + 1_h = 6F_h$
$Y_3 = 7_h \times C_h + 0 \times 2_h + 1A_h = 6E_h$
$Y_4 = 1E_h \times C_h + 3_h \times 2_h + 1_h = 6F_h$

To compute final value, equation (6) is calculated.

$$\text{Final output} = \frac{\sum Y_i \times C_i}{\sum C_i} \qquad (6)$$

$$\text{Final output} = \frac{56 \times 3E + 6F \times 7C + 6E \times 3E + 6F \times 83}{3E + 7C + 3E + 83}$$

Final Value= $6A_h$, Final Value (4-bit) = $A_h = 2.5$ v

Architecture of the project

To implement the idea of that project practically, many components like computing components, Flip-Flops, and block RAMs are needed in FPGA. To represent Mamdani fuzzy rules, four parts of a **Block RAM** are used. Whereas, this Block RAM consists of 10-bit address (represented by merging two labels of the inputs which are 4-bit with 6-bit don't care and 16-bit data), the first two bits in each location represents fuzzy output label as shown in figure 5.

Table (2): Four fuzzy outputs for x_1, x_2 values of the parameters for each fuzzy output

X1	X2	Out1	a_1	a_2	a_3	Out2	a_1	a_2	a_3
0	0	0	0.9524	0.9524	0.9524	2	0.95243	0.95243	1.3955
0	0.5	1.5	0.9524	0.84601	0.84601	0.5	0.9524	0.52946	0.54957
0	1	1	0.95245	0.53632	0.53632	1	0.95245	0.53632	0.4158
0	1.5	0.5	0.95245	0.25385	0.25385	1.5	0.95245	0.76168	0.41143
0	2	2	0.95245	0.81593	0.81593	0	0.95245	0.023134	-0.03445
0	2.5	1.5	0.9524	0.53265	0.53265	0.5	0.9524	0.19768	0.031327
0	3	1	0.9524	0.29905	0.29905	2	0.9524	0.59903	0.20289
0	3.5	0.5	0.95245	0.12747	0.12747	1.5	0.95245	0.12747	0.053871
0	4	2	0.95245	0.11247	0.11247	0	0.85562	0.002042	-0.00817
0.5	0	1.5	0.944	0.11247	0.11247	0	0.94276	0.11247	0.030748
0.5	0.5	1.5	0.92738	0.097703	0.097703	0.5	0.92413	0.094445	-0.00652
0.5	1	1	0.91394	0.074794	0.074794	1	0.91394	0.074794	-0.02689
0.5	1.5	0.5	0.90782	0.057502	0.057502	1.5	0.90782	0.057073	-0.03542
0.5	2	0	0.87802	-0.16727	-0.16727	1.5	1.0199	0.33724	0.18376
0.5	2.5	1.5	0.89138	0.34886	0.34886	0.5	0.92477	0.015862	-0.00148
0.5	3	1	1.09133	0.84866	0.84866	1	1.09133	0.84866	0.33166
0.5	3.5	0.5	0.92418	0.011591	0.011591	1.5	0.96122	0.27085	0.071414
0.5	4	0	0.91382	-0.10838	-0.10838	1.5	0.95729	0.23945	0.063563
1	0	1	0.95638	0.007564	0.007564	0	0.52769	0.007564	-0.39563
1	0.5	1	1.0995	-0.23131	-0.23131	0.5	0.82643	-0.36786	-0.07215
1	1	1	1.2211	-0.45764	-0.45764	1	1.2211	-0.45764	0.32254
1	1.5	0.5	1.2519	-0.71529	-0.71529	1	1.3665	-0.55236	0.46194
1	2	0	1.3	-0.84821	-0.84821	1	1.4855	-0.51725	0.56694
1	2.5	1	1.5224	-0.45841	-0.45841	0.5	1.4618	-0.60995	0.56315
1	3	1	1.5349	-0.39044	-0.39044	1	1.5249	-0.39044	0.63636
1	3.5	0.5	1.5136	-0.46529	-0.46529	1	1.5486	-0.34248	0.65006
1	4	0	1.5026	-0.52665	-0.52665	1	1.5582	-0.30443	0.65958
1.5	0	0.5	0.44812	-0.30443	-0.30443	0	0.13504	-0.30443	-0.25583
1.5	0.5	0.5	0.51739	-0.43166	-0.43166	0.5	0.51739	-0.43166	-0.03427
1.5	1	0.5	0.74342	-0.69378	-0.69378	0.5	0.74342	-0.69378	0.11642
1.5	1.5	0.5	0.85111	-0.64272	-0.64272	0.5	0.85111	-0.64272	0.19821
1.5	2	0	0.85185	-0.73269	-0.73269	0.5	0.9551	-0.59523	0.25754
1.5	2.5	0.5	1.0294	-0.54843	-0.54843	0.5	1.0294	-0.54843	0.38786
1.5	3	0.5	1.0813	-0.48823	-0.48823	0.5	1.0813	-0.48823	0.342
1.5	3.5	0.5	1.1123	-0.43739	-0.43739	0.5	1.1123	-0.43739	0.36236
1.5	4	0	1.1065	-0.50128	-0.50128	0.5	1.1294	-0.39739	0.38042
2	0	0	0.87731	-0.1856	-0.1856	0	0.077311	-0.1856	-0.13765
2	0.5	0	0.1891	-0.55759	-0.55759	0	0.1891	-0.55759	-0.08175
2	1	0	0.29267	-0.55334	-0.55334	0	0.29267	-0.55334	-0.03627
2	1.5	0	0.40267	-0.55379	-0.55379	0	0.40267	-0.55379	0.025184
2	2	0	0.51041	-0.54581	-0.54581	0	0.51041	-0.54581	0.073904
2	2.5	0	0.57161	-0.50109	-0.50109	0	0.57161	-0.50109	0.1095
2	3	0	0.61693	-0.45534	-0.45534	0	0.61693	-0.45534	0.13216
2	3.5	0	0.65782	-0.4195	-0.4195	0	0.65782	-0.4195	0.15261
2	4	0	0.65435	-0.3899	-0.3899	0	0.65435	-0.3899	0.17088
2.5	0	3.5	1.3195	0.12747	0.12747	0	0.11345	0.12747	-0.23173

X1	X2	Out3	a_1	a_2	a_3	Out4	a_1	a_2	a_3
0	0	2	0.95243	0.95243	1.3955	0	0.3486	0.9524	0.9524
0	0.5	2	0.95245	0.86892	1.2285	0	0.27629	0.39282	0.9524
0	1	2	0.95245	0.9304	0.87252	0	0.065176	0.1317	0.95245
0	1.5	2	0.95245	0.9398	0.57057	0	-0.01636	0.055404	0.95245
0	2	2	0.95245	0.81393	0.36091	0	-0.03449	0.023134	0.95245
0	2.5	0	0.9524	0.015178	-0.03768	2	0.238	0.78437	0.9524
0	3	1	0.9524	0.29905	0.10289	2	0.20285	0.55903	0.9524
0	3.5	0	0.95245	0.12747	0.053871	2	0.14285	0.53061	0.95562
0	4	0	0.85562	0.020242	-0.00817	2	0.050122	0.11247	0.95245
0.5	0	2.5	0.84215	0.11247	0.0295.29	0	0.030748	0.11247	0.84276
0.5	0.5	2.5	0.82341	0.093726	-0.00795	0.5	-0.00652	0.084445	0.82413
0.5	1	2.5	0.81297	0.07285	-0.02883	0.5	-0.02903	0.07265	0.81287
0.5	1.5	2.5	0.80767	0.05706	-0.03943	0.5	-0.03813	0.057502	0.80782
0.5	2	0	0.87602	-0.16727	-0.0935	0.5	0.00644	0.020686	0.82573
0.5	2.5	0.5	0.82477	0.015602	-0.00148	0.5	-0.00148	0.015802	0.82477
0.5	3	0.5	0.82481	0.016068	-0.0014	0.5	-0.0014	0.016068	0.82481
0.5	3.5	0.5	0.82418	0.011591	-0.00266	0.5	-0.00265	0.011591	0.82413
0.5	4	2	0.97179	0.35539	0.092548	0.5	0.005592	0.007564	0.82851
1	0	3	1.8171	0.12747	0.18851	0	-0.39563	0.007564	0.82709
1	0.5	3	1.7953	0.1166	0.09677	0.5	-0.07215	-0.36786	0.82643
1	1	3	1.8288	0.15002	0.93015	1	0.32254	-0.45764	1.2211
1	1.5	2.5	1.7103	-0.02771	0.81171	1	0.46184	-0.55236	1.3685
1	2	2	1.6322	-0.18382	0.7336	1	0.56694	-0.51725	1.4655
1	2.5	1	1.5224	-0.45841	0.62381	0.5	0.56315	-0.60995	1.4618
1	3	1	1.5349	-0.39044	0.63636	1	0.63636	-0.39044	1.5345
1	3.5	0.5	1.5136	-0.46529	0.61497	1	0.65006	-0.34245	1.5496
1	4	0	1.5026	-0.52665	0.60402	1	0.65956	-0.30443	1.5582
1.5	0	3.5	1.7314	-0.30443	0.77509	1.5	0.22018	-0.30443	0.89906
1.5	0.5	3.5	1.7553	-0.01835	0.76135	0.5	-0.03427	-0.43166	0.51739
1.5	1	3	1.427	-0.23808	0.57212	1	0.22965	-0.58051	0.51332
1.5	1.5	2.5	1.2844	-0.09939	0.55044	1.5	0.37052	-0.36927	1.1246
1.5	2	2	1.2653	-0.18158	0.46436	1.5	0.39554	-0.31922	1.1621
1.5	2.5	1.5	1.1873	-0.2773	0.41231	0.5	0.38706	-0.54043	1.0294
1.5	3	1	1.143	-0.36578	0.38282	1	0.38282	-0.36578	1.143
1.5	3.5	0.5	1.1123	-0.43739	0.36236	1.5	0.42687	-0.21158	1.2091
1.5	4	0	1.1005	-0.50128	0.35445	1.5	0.43237	-0.1896	1.2173
2	0	4	1.6611	-0.1896	0.65424	0	-0.13765	-0.1896	0.077311
2	0.5	3.5	1.5887	-0.22745	0.57854	0.5	0.01271	-0.51036	0.37802
2	1	3	1.2893	-0.05446	0.46562	1	0.13695	-0.39608	0.62658
2	1.5	2.5	1.0921	-0.03692	0.36977	1.5	0.23205	-0.24343	0.81675
2	2	2	0.9543	-0.10542	0.3011	0.5	0.13446	-0.4387	0.82153
2	2.5	1.5	0.83827	-0.16776	0.24284	0.5	0.15395	-0.33938	0.6605
2	3	1	0.75978	-0.24105	0.20359	1	0.20355	-0.24105	0.75978
2	3.5	0.5	0.71579	-0.31805	0.18159	1.5	0.23956	-0.11515	0.83173
2	4	0	0.69435	-0.3899	0.17088	0	0.17088	-0.3899	0.69435
2.5	0	2.5	0.9744	0.12747	0.062652	2	-0.00609	0.12747	0.80254

To represent the parameters of Linear equation, four parallel parts of another Block RAM are used. Its addresses is represented by merging both crisp inputs with output label (found from Block RAM_1) and its data is divided as shown in figure 6 . Table (3) shows the addresses and data for each part of Block RAM.

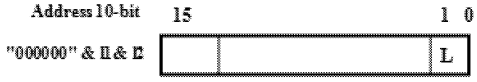


Figure 5: a location of block RAM_1

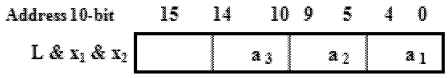


Figure 6: a location of block RAM_2

Each value of parameters shown in table (2) is multiplied by 32 represented by 5- bit in Block RAM. The first two components of the architecture are used to encode crisp inputs (4-bit) into fuzzy inputs (8-bit) depending on the equation (1), (2). Those components operate in parallel (their implementation's period is 1 clock). Then, the next component which consists of four comparators is used to find the minimum degrees (8-bit), its implementation period is 1 clock, and it operates in parallel with Block RAM1 shown in table (2). The outputs of block RAM1 (2-bit per each output) which are Fuzzy output labels found from Fuzzy rules stored as Fuzzy input labels and Fuzzy output labels, are represented as an address for the second block RAM with the crisp inputs. In this state, we needed to delay the values of crisp inputs for two clocks (2 D-FF). Then, next component is Sugeno equation, which consists of four parallel parts, each one is designed to find the outputs of a linear equation (9 bit). Its period is 1 clock. In this component, we needed to delay crisp inputs for three clocks (3 D-FF) as shown in figure 7. To match the operation of this component with minimum degrees values, these values are delayed for two clocks (2 D-FF) to operate in parallel with Sugeno equation components. Minimum degrees (8-bit) and Sugeno outputs (9- bit) enter to the last components in parallel to compute the final value. Last component consists of multipliers, adders, and divider to compute Equation (5). After, implementing the architecture using FPGA, the practical properties of the architecture are shown in table (4). Where as, Number of consumed slices and used BRAMs are 8 % and 50 % respectively.

Table (3): Addresses of second Block RAM

	X1	X2	L1	L2	L3	L4	Add1	Add2	Add3	Add4
	Hex	Hex					RAM1	RAM2	RAM3	RAM4
1	0	0	0	0	2	1	0	0	200	100
2	0	2	0	0	2	1	2	2	202	100
3	0	4	0	0	2	1	4	4	204	100
4	0	6	0	0	2	0	6	6	206	6
5	0	8	0	0	2	1	8	8	208	108
6	0	A	0	0	1	0	A	A	10A	A
7	0	C	0	0	1	0	C	C	10C	C
8	0	E	0	0	1	0	E	E	10E	E
9	0	F	0	0	1	0	F	F	10F	F
10	2	0	0	0	2	1	20	20	220	120
11	2	2	0	0	2	1	22	22	222	122
12	2	4	0	0	2	1	24	24	224	124
13	2	6	0	0	2	1	26	26	226	126
14	2	8	0	0	2	1	28	28	228	128
15	2	A	0	0	1	0	2A	2A	12A	2A
16	2	C	0	0	1	0	2C	2C	12C	2C
17	2	E	0	0	1	0	2E	2E	12E	2E
18	2	F	0	0	1	0	2F	2F	12F	2F
19	4	0	0	0	2	1	40	40	240	140
20	4	2	0	0	2	1	42	42	242	142
21	4	4	0	0	2	1	44	44	244	144
22	4	6	0	0	2	1	46	46	246	146
23	4	8	0	1	2	1	48	148	248	148
24	4	A	0	0	1	0	4A	4A	14A	4A
25	4	C	0	0	1	0	4C	4C	14C	4C
26	4	E	0	0	1	0	4E	4E	14E	4E
27	4	F	0	0	1	0	4F	4F	14F	4F
28	6	0	0	0	2	1	60	60	260	160
29	6	2	0	0	2	1	62	62	262	162
30	6	4	0	0	2	1	64	64	264	164
31	6	6	0	0	2	1	66	66	266	166
32	6	8	0	0	2	1	68	68	268	168
33	6	A	0	0	1	0	6A	6A	16A	6A
34	6	C	0	0	1	0	6C	6C	16C	6C
35	6	E	0	0	1	0	6E	6E	16E	6E
36	6	F	0	0	1	0	6F	6F	16F	6F
37	8	0	0	0	2	1	80	80	280	180
38	8	2	0	0	2	1	82	82	282	182
39	8	4	0	0	2	1	84	84	284	184
40	8	6	0	0	2	1	86	86	286	186
41	8	8	0	0	2	1	88	88	288	188
42	8	A	0	0	1	0	8A	8A	18A	8A
43	8	C	0	0	1	0	8C	8C	18C	8C
44	8	E	0	0	1	0	8E	8E	18E	8E
45	8	F	0	0	1	0	8F	8F	18F	8F
46	A	0	2	0	2	2	2A0	A0	2A0	2A0
47	A	2	2	1	2	2	2A2	1A2	2A2	2A2
48	A	4	2	1	2	2	2A4	1A4	2A4	2A4
49	A	6	2	1	2	2	2A6	1A6	2A6	2A6
50	A	8	2	1	2	2	2A8	1A8	2A8	2A8
51	A	A	1	0	2	2	1AA	AA	2AA	2AA

CONCLUSION

In this paper, we tried to present the possibility to implement Fuzzy Systems depending on Mamdani rules into Sugeno Controller design. Despite Mamdani Fuzzy Controllers are more famous than Sugeno Controllers, but digital hardware design of Mamdani Controllers is longer and more complex than the second one. Where as, Mamdani design consists of several parts for computing fuzzy outputs. These parts contain number of dividers which consume

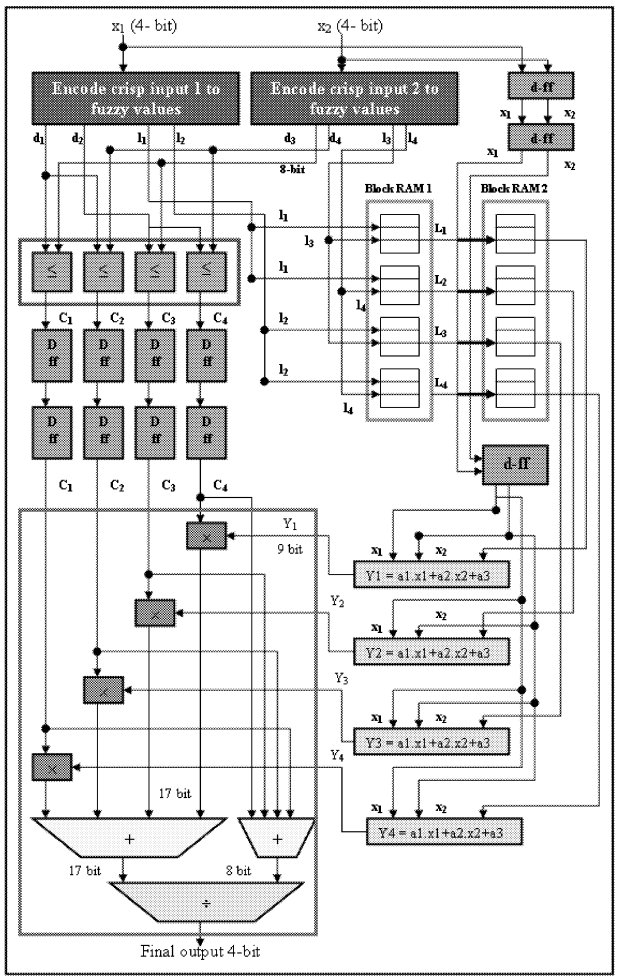


Figure 7: FPGA Architecture

large area of FPGA in addition to delay in implementation. On the other hand, Sugeno design has many advantages as working well with linear techniques (as PID), having guaranteed continuity of the output surface, and working well with optimization and adaptive techniques, in addition, Sugeno design distinguishes by consuming small area of FPGA for implementing linear equations with fast computation. From this concept, design an approach to convert Fuzzy rules into linear rules was implemented. That has given the ability to implement Mamdani Fuzzy Control systems into Sugeno design, which has been implemented using Supervised Neuron. Figures (8-10) shows the results of architecture. Figure 8 illustrates the fuzzy obtained values from converting crisp inputs into fuzzy values. Whereas, each crisp input gives two fuzzy values with its labels as result to overlapping between memberships. Then, by selecting minimum two fuzzy values, minimum degrees are obtained and shown in figure 9. Figure 10 shows Sugeno values after converting Mamdani values depending Single Perceptron Neuron, the relationship between fuzzy outputs and Sugeno outputs, which was represented by weights and biases, was found. By comparing final output of practical implementing shown in figure 11 with theoretical output computed previously and output obtained from implementing the application in Matlab, the output would be almost equal.

Practical final output = A_h = 2.5 v,
Theoretical final output = 2.5 v
Matlab final output = 2.45 v

Table (4): Practical properties of FPGA architecture

Logic utilization	Used	Available	Utilization
No. of Slices	287	3584	8 %
No. of Slice Flip Flops	466	7168	6 %
No. of bonded IOBs	13	141	9 %
No. of BRAMs	8	16	50 %
No. of GCLKs	1	8	12 %

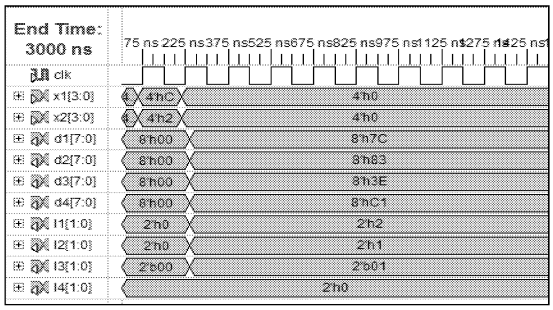


Figure 8: Conversion two crisp inputs into Fuzzy Values

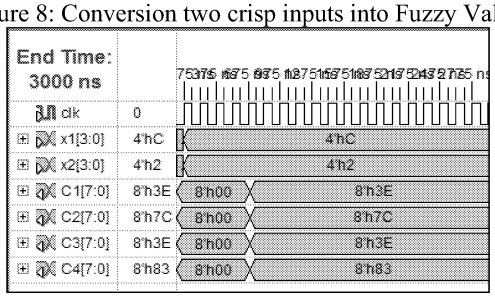


Figure 9: Four Minimum degrees

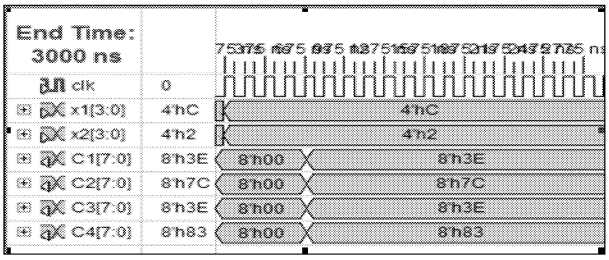


Figure 10: Four Sugeno Outputs

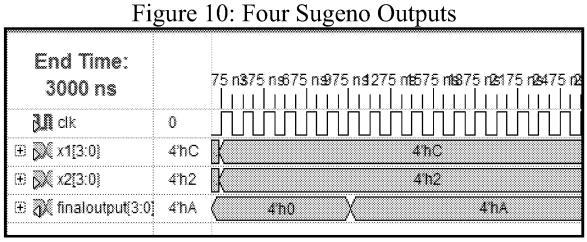


Figure 11: Practical Final Value

REFERENCES

Martin T. Hagan et al. 2002 **"Neural Network Design"** ISBN: 7-111-10841-8. China Machine Press.

S. Singh and K. S. Rattan, 2003 **"Implementation of a fuzzy logic controller on an FPGA using VHDL,"** 22nd Int. Conf. of the North American Fuzzy Information Processing Society, NAFIPS 2003, pp. 110 - 115, July 24-26, 2003.

V. GEORGESCU, 2002 **"Capturing and Tuning Nonlinear Characteristics of Economic Stabilization Systems by Fuzzy Control Techniques".** Computational Economics 19: 247–271, 2002. ©2002 Kluwer Academic Publishers. Printed in the Netherlands.

Jan Jantzen, 1998 **"Design of Fuzzy Controllers"**. Technical University of Denmark, Department of

Automation, Bldg 326, DK-2800 Lyngby, Denmark. Tech. report no 98-E 864 (design), 19 Aug 1998.
Yen and R. Langari,, 2003 **"Fuzzy Logic Intelligence, Control, and Information"**, Singapore, Pearson Education, 2003.
V. Salputra, 2000**"A Fuzzy RISC Processor"** IEEE Trans. on Fuzzy Systems, vol. 8, Issue: 6, pp. 781 - 790, Dec.2000.

COMMUNICATION NETWORKS SIMULATION

Design and Implementation of Computer Networks Based on ATM Technology

Dr.Saad ALshaban, Lubna K. M. Al-Azzawi
Dr. Sattar Sadkhan
Computer Eng. Dept. of University of Technology / Iraq
salshabaan@yahoo.com

KEYWORDS
ATM, LAN emulation,, cell switching, Switch model, Computer networks.

ABSTRACT

This paper is concerned with designing and simulation of computer networks based on Asynchronous Transfer Mode (ATM) Technology which is with a high-performance; cell-oriented switching technology that utilizes fixed small size packets to carry different types of traffic. The work presents principles of operation, protocols and internetworking with ATM technology.
There is increasing interest for internetworking with ATM technology in the Local Area Networks (LANs) environments to achieve a better efficient communication, henceforth the researcher set a design of internetworking using ATM technology (ATM LANE) to connect the LANs in the two buildings of the Computer and Software Engineering Department in the University of Technology as a case study. Simulation program using Visual Basic language and Matlab program is depended on the computer as a delivery system to all activities of the simulation program. The paper has achieved the following important conclusions from internetworking with ATM technology.
1. It enables users to acquire the desired level and quality of service because the transmission time delay using ATM technology is very limited.
2. It offers guaranteed network access for voice and video applications because the throughput rate in the ATM LANE for different multimedia communication messages is high with hundreds of Mbps

INTRODUCTION

The ATM offers flexible bandwidth allocation at high speeds and a scalable architecture that is not limited by this technology. ATM is the first technology to support integration across local, metropolitan, and wide area networks. While ATM offers many benefits in the LAN environment, ATM LANs must provide transparent internetworking with existing LAN technologies [Davis, an]

The ATM is used in a LAN environment primarily as a backbone for interconnecting with legacy LANs that is, the old-fashioned but paid-for pre-ATM equipment such as Token-Ring and Ethernet [Ian].The use of ATM switches as a backbone for connecting legacy LAN traffic results in a series of incompatibilities that must be overcome[Preston]. The major incompatibilities between ATM and legacy LANs include their connection method, address, and transport mechanism [7]. An ATM network uses a connection-oriented switching mechanism to route traffic in the form of 53-byte cells from source to destination using Virtual Path Identifiers (VPIs) and Virtual Channel Identifiers (VCIs) [8].

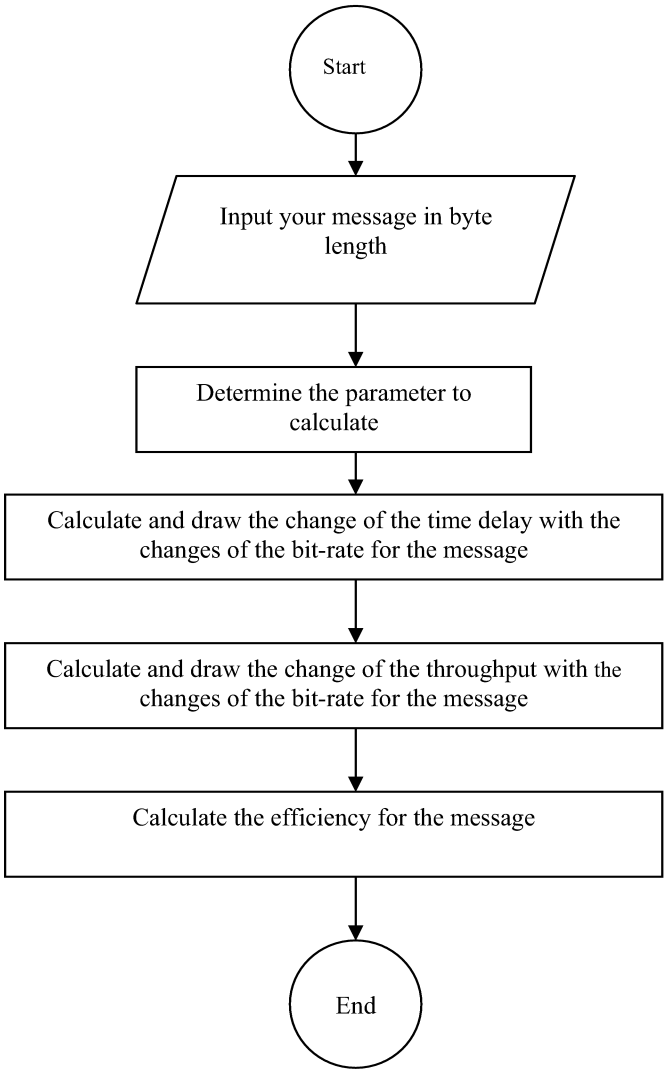


Figure [1] shows general design for present computer network

In comparison, the LAN standards were published by the IEEE: 802.3 for the Ethernet and 802.5 for the Token Ring network are connectionless, user applications and protocols are not required to establish a connection before submitting data for transmission over the LAN media [Kolluri], transmitting variable length frames containing 48-bit media access control (MAC) addresses to all stations on a network [Gordon, John].To overcome these incompatibilities, the ATM Forum developed a specification referred to as LAN Emulation (LANE)[James]

LIMITATION OF THE RESEARCH

- To computer program (Matlab Version7.0) has been used at present time:
a) Calculate and draw the graphics of transmission time delay.
b) Calculate and draw the graphics of throughput.
c) Find the efficiency of the transmission using ATM technology.
- To set a program which
is capable of calculating and to draw the parameters mentioned above for any kind of message set by the user.
- Designing and implementing a computer aided learning (The Microsoft office XP 2002).
- The use of (Adobe Photo Shop Version 6.0) program to best (CAL) simulation program using the ssue the computer aided learning (CAL) simulation program

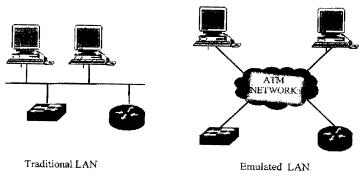


Figure [2] ATM Network can emulation Physical LAN

ATM LAN Emulation designs are evolving rapidly with deployment of switching, three models have emerged as generic designs, these models are:
A- Scaled Switching Model
B- Distributed Routing Model
C- Distributed Multilayer Switches Model[ATM Forum].

SCALED SWITCHING MODEL

LAN Emulation (LANE)
The LAN Emulation (LANE) specification is standardized by ***ATM*** The is interconnection LANs and ATM networks, the network becomes a mix of two types of technologies: connectionless LAN (the broadcast mode operation of a legacy LAN) and connection oriented of the ATM technology [James]

LAN Emulation is designed to allow existing legacy LANs network applications and network protocols to run over ATM networks . Since one of the goals of the ATM is to provide complete worldwide connectivity, it is important to allow multiple LANs to exist on the same physically interconnected the single ATM network [ATM Forum]. It uses ATM as a backbone for connecting "legacy" networks. It's also designed to support both of directly attached ATM end systems and end systems attached through edge devices. Figure [1] sows a traditional LAN and an Emulated LAN (ELAN) [ATM Forum]

Broadcast and Unknown Server (BUS)

The BUS acts as a central point for distributing broadcasts and multicasts. ATM is essentially a point-to-point technology without "any-to-any" or "broadcast" support [60]. LANE solves his problem by centralizing the broadcast support in the BUS.Each (LAN emulation Client) LEC must set up a multicast send (Virtual circuit connection) VCC to the BUS, then adds the LEC as a leaf to its point-to-multipoint VCC known as the *multicast forward VCC)* [Locken].

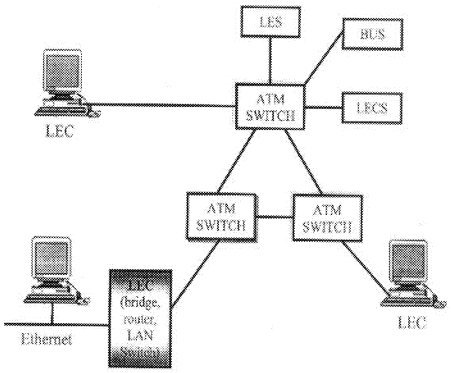


Fig ure [3} LAN Emulation Components

design of scaled switching model deploys switching at all levels of the design without the use of routers. In this design, each layer consists of switches [A TM Forum]. ATM LANE permits multiple LANs to exist over a single switched ATM.

Distributed Routing Model

In a large ATM LAN Emulation, routers (layer 3 switching) add to ATM LANE. The routers act as a LAN Emulation Clines (LECs) , The LANE Configuration Server (LECS), LANE Server (LES), and Broadcast and Unknown Server (BUS) for ELAN are configured as the usual way in the ATM switch, The distributed routing model deploys LAN Switches (layer 2 switching) in the access layer and routers in the distribution layer [Saad Alshaban].

Distributed Multilayer Switches Model

The final model of the large ATM LAN Emulation is involved the multilayer switches with ATM LANE backbone [Saad Alshaban] .Figure 4 shows a general flowchart network design as shown below.

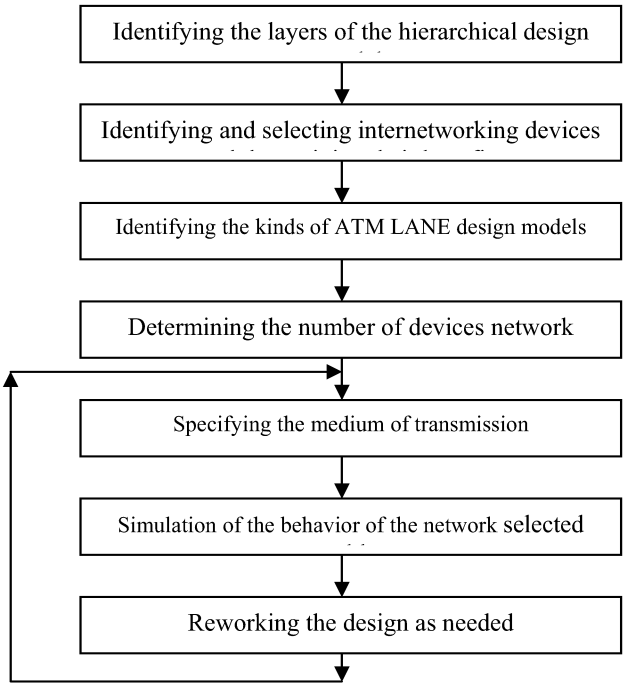


Figure [4] General Network Design procedure

.

In order to achieve a better learning experience and improve administrative efficiencies for the researcher at present adopted the first model "Scaled *Switching Model*" to design the ATM LANE for the department because scaled switching model is a low-cost and easy-to-install solution for a small ATM LANE and it is easy to manage.The Department consists of two buildings (A)and (B). Since the two buildings are close together in a small area, and the number of LANs is few, therefore there is no need for a separate device in the distribution layer, the core and distribution layers are present in a single entity, the ATM switch (example: Light Stream® 1010 switch.

DISCUSSION OF THE RESULTS

l Network Design Process

The following outline discusses the several area that should carefully be considered when planning internetworking implementation [Cisco System]. Fig. 4 Shows a general network design procedure which should be used in designing the ATM LANE, While Fig.5 shows the calculation program for time delay and for through put efficiency. Fig 6 shows Design ATM LANE Emulation to the Department.

According to the theoretical background and calculation results obtain the following Based on the result of the efficiency calculation, it is obvious that the ATM LLANE allows for the integration of networks improving high efficiency and ma flexibility and nagging ability. It provides a good bandwidth can be used efficiently from desktop computers to local area networks.- A network based on ATM technology provides a possibility to offer a variety of communication speeds and variable bit rates for a variety of media, with the right kind of communications quality for every application.- Advances in fiber optic have led to the emergence of very high speed networks based on asynchronous transfer mode ATM. The ATM interfaces can span ,the range from 155.52 Mbps to 622.08 Mbps.

Role of CAL

The role of the computer aided learning (CAL) simulation program proved a better learning achievement in relevance to acquire the same subject technology that easily scaled from (155 Mbps), (622 Mbps),(2.488Gbps), and up to (10Gpbs). Deploying LAN switches within wiring closets with increase the demand for backbone and wide area bandwidth to support higher speeds, ATM switching will be ideal for this role providing , the orders of magnitude Access bandwidths .his flexibility and scalability assures that ATM will be around for a long time. ATM technology is a cell-based communications network has designed from

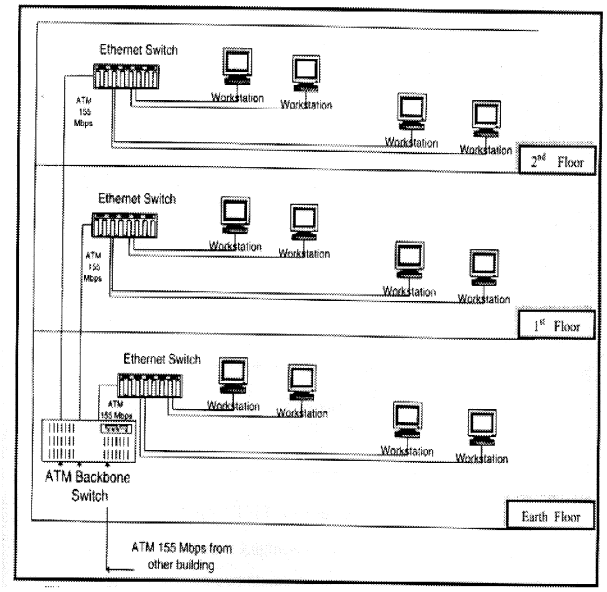


Figure 5. Design ATM LAN Emulation to the Department

the onset to be scaleable and flexible in Geographic distance .Number of user intensive applications across networks. In order to achieve a better learning experience and improve administrative efficiencies for the researcher at present adopted the first model "Scaled *Switching Model*" to design the ATM LANE for theDepartment as shown in fig 6, because scaled switching model is a low-cost and easy-to-install solution for a small ATM LANE and it is easy to manage.

CONCLUSIONS

It can be delivered from the results obtained:

The following conclusions, Asynchronous Transfer Model (ATM) is the best computer internetworking technology designed to carry video, voice and data traffic simultaneously. ATM technology provides the level of service necessary to run each of these bandwidths.

FUTURE WORK

It is suggested here to use FDDI in networking purposes in order to develop fiber optics to speed Up the proposed communications between above Department buildings even as it costs a lot.

REFERENCES

ATM Forum, " LAN Emulation Sample Configuration" , http:// www. aim forum, corn, 2002 .

ATM Forum, "An Overviews/ATM LAN Emulation", http://www .aim forum, corn, 2002.

ATM Forum, "ATM State of The Art-LAN Emulation", http ://www.atmforum.com, 2001.

ATM Forum, ATM LAN Design" http://www.atmforum.com,2002.

ATMForum, "Designing Switched LAN Internetworks", http://www.atmforum.com, 2002 .Wittiam, S. , "Data & computer communication". Prentice –Hall Inc, 2000.

Davis, R. H., "ATM for Public Networks", Me Graw-Hall,2002 [2] Andrew, C., "LAN Emulation (LANE)", http://www.ibm. corn, 2002.

Dickey, M., " Routing in Today's Internet-Works", Van Nostrand Reinnold, 2001.

Gordon, B., " Computers in Communication" , Mcgraw Hill Publishing Company, 1997.

Ian, A. F., and Keith, L. B., "ATM Local Area Networks", IEEE communication magazine, July, 1997.

IBM System, "Evolution of local Area Networking", International Technical Support Organization, 1999 .

James, E. G. & Phillip, T. R., "Local Area networks: A Business-Oriented Approach" , John Wiley & Sons, Inc, 2000.

John, P. S., "Handbook of Local Area Networks", CRC Press, 1998

Kolluri, S.; and Others, "Digital Network Design", Elsevier CPE319, 2001.

Locken, H. A., "Asynchronous Transfer Mode Fundamentals", Galina pildush published, 2002.

Preston, G., "Internetworking Technologies handbook", Publisher Macmillan Computer Publishing, 1999.

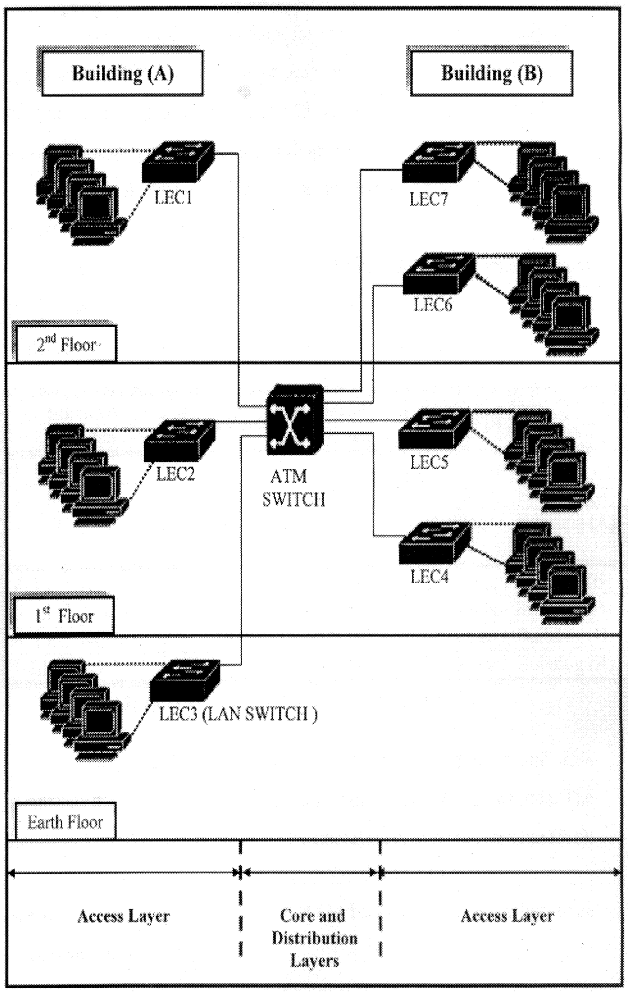


Figure (6): Design ATM LANE Emulation to the Department

o- End Network for Delivering Broadband in Higher Education Environments", http://www. cisco. system, 2002.

Lubna Alazzawi " Design and Simulation of computer networks based on ATM Technology " M.Sc Thesis submitted to the university of technology,University Baghdad, Iraq.

VERIFICATION OF SQUARE COMMUNICATION GRID PROTOCOLS VIA INFINITE PETRI NETS

Shmeleva T.R., Zaitsev D.A., Zaitsev I.D.
Odessa National Academy of Telecommunications
Kovalska, 1, Odessa 65029, Ukraine
E-mail: tishtri@rambler.ru, zsoftua@yahoo.com, ivan_d_zaitsev@mail.ru

KEYWORDS
Communication grid, Protocol, Verification, Infinite Petri Net, Deadlock

ABSTRACT

A technique of the linear invariants calculation for infinite Petri nets with the regular structure is presented and studied on the example of square communication grids of an arbitrary size. It is proven that the compulsory buffering of the packets inevitably leads to possible blockings of communicating devices. The structure of complex deadlocks involving an arbitrary number of communicating devices caused by both the cycle of blockings and the isolation is studied. *

INTRODUCTION

The Petri net application for the verification of telecommunication protocols is a rather traditional direction of research (Berthelot and Terrat 1982; Diaz 1982). The majority of known works study communication processes in pairs of communicating devices. But anomalies may occur which involve an arbitrary number of communicating devices and the present paper proves this statement. As the number of communicating devices and the structure of the network are considerably varied for real-life networks, a technique is required that could manage an arbitrary number of devices constituting an arbitrary structure. Recently the parametric composition of functional Petri nets (Zaitsev 2005) was applied (Zaitsev and Zaitsev 2006) for the analysis of infinite linear structures of communicating devices (Marsan, A.M.; Chiola, G. and Fumagalli, A. 1987). A simpler direct approach (Shmeleva 2007) was applied for treelike infinite structures. The present work is restricted to the square communication grid (matrix) infinite structure of communicating devices but it seems that the obtained results might be generalized for an arbitrary structure as well.

THE MODELS CONSTRUCTION

For the composition of infinite communication structures, submodels of a typical communication device and a terminal device are required. Further we compose such a regular communication structure as a square matrix of communicating devices of an arbitrary size.

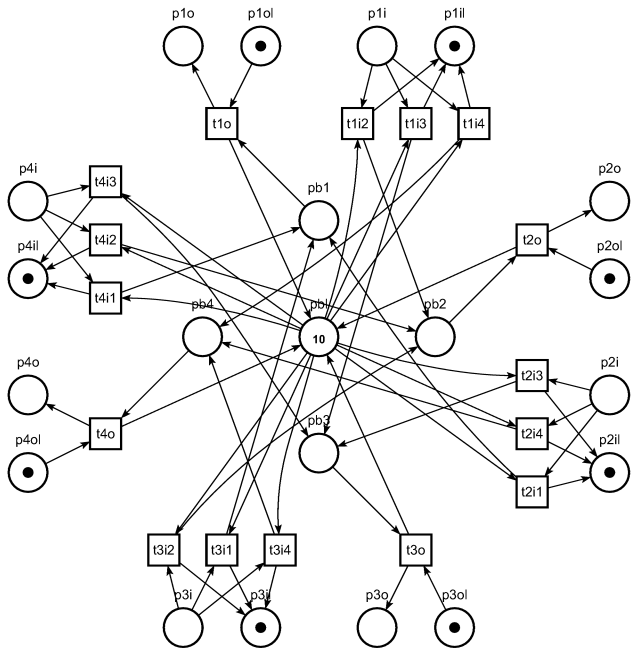
The Model of a Communication Device

Let us consider such real-life communication devices as switches and routers, for instance, Ethernet switches and IP/MPLS routers. Their basic function (Russell 2004) is the redirection of the arrived packets to the destination port. So the model of a communication device consists of ports models. Usually each port works in the full-duplex mode that allows simultaneous transmission in both directions. This is provided by the different channels of the port: the input channel and the output channel. A channel of a port has its internal buffer for the allocation of one packet. Moreover, the communication device uses an internal buffer with the limited capacity where it stores the packets before they are put into the destination port. The usage of the switching and routing tables is not modeled but the redirection function is represented by the allocation of the arrived packet for each possible destination. This abstract description suits the routers operation with the compulsory buffering of the packets and without the cut-through possibilities (Russell 2004).

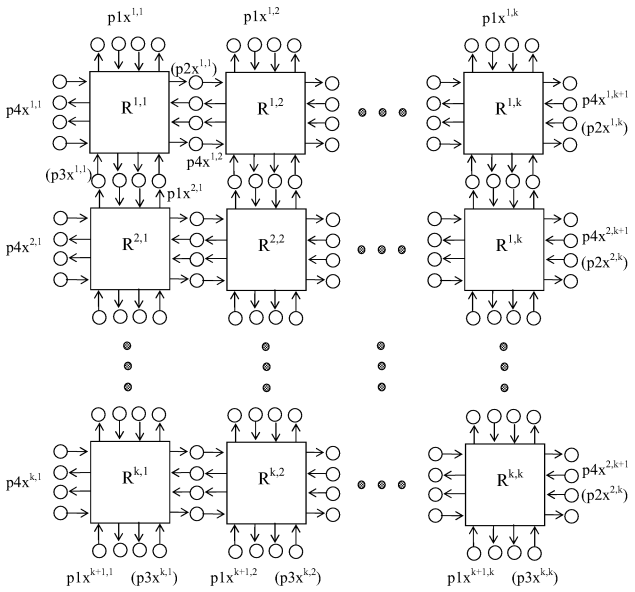
The constructed model of a communication device is represented in Fig. 1a. It has four ports for the further composition of the communication matrix but it might be constructed with an arbitrary number of ports as well. All the ports are situated on the sides of a square and are numbered clock-wise from 1 (the upper side) to 4 (the left side). A port consists of two channels: input (i) and output (o). Each channel is represented by a pair of places: one for the packet allocation and the other for modeling the port buffer capacity which is equal to 1. For instance, port 1 is modeled by the following places: $p1i$ - input buffer; $p1il$ - limitation of the input buffer capacity (equal to 1); $p1o$ - output buffer; $p1ol$ - limitation of the output buffer capacity (equal to 1). The internal buffer of the communication device is modeled by the separate places where the packets with the corresponding destinations are allocated: $pb1$, $pb2$, $pb3$, $pb4$. For instance, place $pb1$ stores the packets redirected to port 1. Moreover, the capacity of the internal buffer is modeled by place pbl. The operation of the port output channel is represented by the only transition $t*o$, for instance, $t1o$ for port 1. The transition checks the availability of the port buffer $p1ol$,

* Work supported by NATO grant ICS CLG 982698

gets a packet from the corresponding place $pb1$, puts the packet into the port output buffer $p1o$ and increases the capacity of the internal buffer pbl. The operation of the port input channel is more sophisticated because it models the process of the destination choice. For each port it is modeled by three transitions (3=4-1). For instance, $t1i2$, $t1i3$, $t1i4$ for port 1. Transition $t1i2$ allocates the packets redirected from port 1 to port 2 into the internal buffer place $pb2$ and so on.



a) Model of cell (*n2s1*)



b) Grid composition (*n2sk*)

Figures 1: Model of square grid communication structure

Notice that the constructed model constitutes a definite balance between the complexity of real-life devices and the possibilities for formal analysis. At least, the essential features (Russell 2004) are modeled: redirection of the packets and their intermediate allocation into the internal buffer with the limited capacity.

The Model of a Communication Structure

The matrix (two dimensions) structure of communication devices is studied in the present paper. As the real-life communication structure may consist of an arbitrary number of communication devices, special methods should be developed that handle the infinite number of communication devices. Furthermore, it will be shown that the communication structure brings us anomalies which can not be found during the traditional study of communication processes in pairs of devices (Berthelot and Terrat 1982; Diaz 1982). The results are obtained for the regular structures such as the matrix but it seems they might be generalized on arbitrary structures.

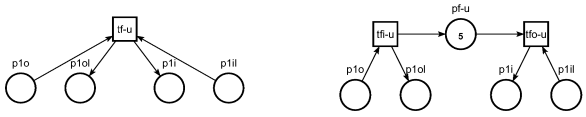
Notice that the constructed model *n2s1* is a functional Petri net (Zaitsev 2005). Let us construct a communication structure via the composition of communication device models situated in the cells of a matrix. Square matrices of the size k are studied, where k is an arbitrary natural number. So each device $R^{i,j}$ in the matrix is defined by two indices: i – for vertical direction and j – for horizontal, $i = \overline{1,k}$, $j = \overline{1,k}$. The matrix communication structure is represented in Fig. 1b.

The connection of communication devices is provided by the fusion (union) of corresponding contact places. For instance, for an internal communication device $R^{i,j}$, $i = \overline{2,k-1}$, $j = \overline{2,k-1}$, the places of port 1 are fused with the corresponding places of port 3 for the device $R^{i-1,j}$ in such a way that place $p_{1o}^{i,j}$ is fused with $p_{3i}^{i-1,j}$, place $p_{1ol}^{i,j}$ – with $p_{3il}^{i-1,j}$, place $p_{1i}^{i,j}$ – with $p_{3o}^{i-1,j}$, place $p_{1il}^{i,j}$ – with $p_{3ol}^{i-1,j}$. So the full-duplex mode of communication via two channels of the ports is modeled. The rules of device $R^{i,j}$ connection may be formulated in the following way: the upper side – port 1 to port 3, device $(i-1, j)$; the right side – port 2 to port 4, device $(i, j+1)$; the bottom side – port 3 to port 1, device $(i+1, j)$; the left side – port 4 to port 1, device $(i, j-1)$. After the composition, the contact places have duplicate names. To avoid duplicity, the names of the places for the ports 1, 4 (the upper and left sides) will be considered with respect to the current device, for the ports 2, 3 – with respect to the neighbor devices and their ports 1, 4 correspondingly. So the names of the fusion places have only the prefixes of the ports 1, 4. Moreover, to simplify further notations, the places of the right and bottom borders of the matrix are named with respect to non existing devices with the indices equaling to $k+1$. So the numbers

of the ports 2 and 3 do not appear in the matrix. An example of the communication matrix (with attached terminal devices) for $k = 2$ is represented in Fig. 3.

The Models of Terminal Devices

The communication devices may be attached to each other constituting a communication structure but they are created only for the packets transmission among the terminal devices: workstations and servers. In the present work, the client-server technique of interconnection is not studied, so the types of terminal devices are not distinguished as in (Zaitsev and Shmeleva 2006.). An abstract terminal device provides at least two basic functions: send packet and receive packet. These basic functions are provided by the models represented in Fig. 2.



a) a simple reflection of packets (*n0f*)

b) with the buffer of the packets (*nf*)

Figures 2: Petri net models of a terminal device

The Fig. 2a gives the simplest model that only reflects the arrived packets from the input to the output via transition $tf-u$; names of the places are given with respect to the places of a communication device port. So the model of a terminal device may be attached by the fusion of the places with the same names. The model in Fig. 2b contains an internal buffer of the packets $pf-u$; transition $tfi-u$ models the input of the packets, while transition $tfo-u$ models the output. The suffix $-u$ means the upper row of terminal devices that supposes port 1 for attachment; the left, right and bottom (down) rows contains the suffices $-l$, $-r$, $-d$ correspondingly. An example of the communication matrix with attached terminal devices (type a) is represented in Fig. 3.

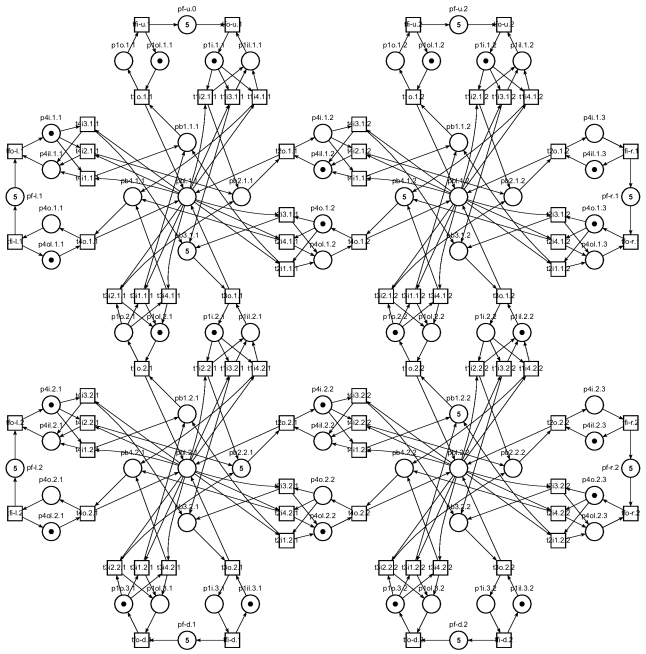
CALCULATING P-INVARIANTS

The described composition of the model allows the application of the technique for Petri net analysis via the composition of its functional subnets (Zaitsev 2005). This approach was applied for the Ethernet protocols with the bus structure verification for an arbitrary number of devices on the bus (line structure) (Zaitsev and Zaitsev 2006). But it brings us the explosion of solutions basis for the composition system and hinders the application of this technique. A rather simple technique of the direct construction of an infinite linear system for p- and t-invariants applied for the switched Ethernet protocols verification (binary tree structure) (Shmeleva 2007) seems more adequate.

The following infinite linear system of equation is constructed for p-invariants calculation of Petri net *n2sk* (Fig. 1b) – the matrix of k devices *n2s1* (Fig. 1a), where k is an arbitrary natural number:

$$\begin{cases}
t_{1o}^{i,j} : x_{1ol}^{i,j} + x_{b1}^{i,j} = x_{1o}^{i,j} + x_{bl}^{i,j}, \\
t_{1i2}^{i,j} : x_{1i}^{i,j} + x_{bl}^{i,j} = x_{1il}^{i,j} + x_{b2}^{i,j}, \\
t_{1i3}^{i,j} : x_{1i}^{i,j} + x_{bl}^{i,j} = x_{1il}^{i,j} + x_{b3}^{i,j}, \\
t_{1i4}^{i,j} : x_{1i}^{i,j} + x_{bl}^{i,j} = x_{1il}^{i,j} + x_{b4}^{i,j}, \\
t_{4o}^{i,j} : x_{4ol}^{i,j} + x_{b4}^{i,j} = x_{4o}^{i,j} + x_{bl}^{i,j}, \\
t_{4i1}^{i,j} : x_{4i}^{i,j} + x_{bl}^{i,j} = x_{4il}^{i,j} + x_{b1}^{i,j}, \\
t_{4i2}^{i,j} : x_{4i}^{i,j} + x_{bl}^{i,j} = x_{4il}^{i,j} + x_{b2}^{i,j}, \\
t_{4i3}^{i,j} : x_{4i}^{i,j} + x_{bl}^{i,j} = x_{4il}^{i,j} + x_{b3}^{i,j}, \\
t_{2o}^{i,j} : x_{4il}^{i,j+1} + x_{b2}^{i,j} = x_{4i}^{i,j+1} + x_{bl}^{i,j}, \\
t_{2i1}^{i,j} : x_{4o}^{i,j+1} + x_{bl}^{i,j} = x_{4ol}^{i,j+1} + x_{b1}^{i,j}, \\
t_{2i3}^{i,j} : x_{4o}^{i,j+1} + x_{bl}^{i,j} = x_{4ol}^{i,j+1} + x_{b3}^{i,j}, \\
t_{2i4}^{i,j} : x_{4o}^{i,j+1} + x_{bl}^{i,j} = x_{4ol}^{i,j+1} + x_{b4}^{i,j}, \\
t_{3o}^{i,j} : x_{1il}^{i+1,j} + x_{b3}^{i,j} = x_{1i}^{i+1,j} + x_{bl}^{i,j}, \\
t_{3i1}^{i,j} : x_{1o}^{i+1,j} + x_{bl}^{i,j} = x_{1ol}^{i+1,j} + x_{b1}^{i,j}, \\
t_{3i2}^{i,j} : x_{1o}^{i+1,j} + x_{bl}^{i,j} = x_{1ol}^{i+1,j} + x_{b2}^{i,j}, \\
t_{3i4}^{i,j} : x_{1o}^{i+1,j} + x_{bl}^{i,j} = x_{1ol}^{i+1,j} + x_{b4}^{i,j}, \quad i = \overline{1,k}, \quad j = \overline{1,k}.
\end{cases} \tag{1}$$

Notice that the standard technique of invariants calculation (Murata 1989) is applied directly to the infinite Petri net. In the system for p-invariants calculation each equation corresponds to a transition; the sums of variables for the input and output places are equal. It is easy to check that each transition of the net *n2dk* was considered in the system (1).



Figures 3: Petri net model of a grid with attached terminal devices for $k = 2$ (*nf2s2*)

The first 4 equations correspond to the transitions of ports 1, the next 4 – to the transitions of ports 4. They use variables of the places with the indices of the current device (i, j), according to the naming rules. The next 4 equations describe the transitions of ports 2; the last 4 equations – the transitions of ports 3. They use the variables of ports 1, 4 places with the indices of the neighbor device $(i+1, j)$, $(i, j+1)$ correspondingly, instead of ports 3, 2 places with the indices of the current device (i, j), according to the naming rules. Let us count the number of the equations and variables in the system (1). The total number of equations (transitions): $N_k^{n2t} = 16 \cdot k^2$. The total number of variables (places): $N_k^{n2p} = 13 \cdot k^2 + 8 \cdot k$.

The universal methods for the infinite systems of the linear equations under the rings (integer numbers) solving, especially into semigroups (nonnegative integer numbers) are unknown. We applied a heuristic method of a general solution construction in the parametric form. The general solution of the system (1) may be represented as:

$$\left(\begin{array}{l}
(p_{1i}^{i,j}, p_{1il}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k+1};\\
(p_{1o}^{i,j}, p_{1ol}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k+1};\\
(p_{4i}^{i,j}, p_{4il}^{i,j}),\ i=\overline{1,k+1},\ j=\overline{1,k};\\
(p_{4o}^{i,j}, p_{4ol}^{i,j}),\ i=\overline{1,k+1},\ j=\overline{1,k};\\
(p_{b1}^{i,j}, p_{b2}^{i,j}, p_{b3}^{i,j}, p_{b4}^{i,j}, p_{bl}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k};\\
(((p_{1i}^{i,j}, p_{1o}^{i,j}, p_{4i}^{i,j}, p_{4o}^{i,j}, p_{b1}^{i,j}, p_{b2}^{i,j}, p_{b3}^{i,j}, p_{b4}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k}),\\
\qquad ((p_{4i}^{i,k+1}, p_{4o}^{i,k+1}),\ i=\overline{1,k}),\ ((p_{1i}^{k+1,j}, p_{1o}^{k+1,j}),\ j=\overline{1,k}))\\
(((p_{1il}^{i,j}, p_{1ol}^{i,j}, p_{4il}^{i,j}, p_{4ol}^{i,j}, p_{bl}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k}),\\
\qquad ((p_{4il}^{i,k+1}, p_{4ol}^{i,k+1}),\ i=\overline{1,k}),\ ((p_{1il}^{k+1,j}, p_{1ol}^{k+1,j}),\ j=\overline{1,k}))
\end{array}\right) \quad (2)$$

The way of the solutions description is common enough for sparse vectors and especially for the Petri net theory. Only nonzero components are mentioned by the name of a corresponding place. The nonzero multiplier 1 is omitted; in case it is not the unit, the notation $p*x$ is used where x is the value of the invariant for place p. Such notation is adopted in the Tina software (Berthomieu et el. 2004) which was used for obtaining the Petri net figures in this paper. A line of the matrix (2) gives us a set of lines according to the used indices i and j except the last two lines which contain variable number of components given by indices. The total number of solutions is $N_k^{n2pinv} = 5 \cdot k^2 + 4 \cdot k + 2$.

We did not manage to prove that the matrix (2) is the basis of nonzero solutions of the system (1) but it is possible to ground that each line of (2) is a solution of (1). And this fact allows the proof of p-invariance for the net *n2sk*.

Lemma 1. Each line of the matrix (2) is a solution of the system (1).

Proof. Let us substitute each parametric line of (2) into each parametric equation of the system (1). It gives us the correct statement. For instance, let us substitute the first line of (2)

$(p_{1i}^{i',j'}, p_{1il}^{i',j'}),\ i'=\overline{1,k},\ j'=\overline{1,k+1};$

into the second equation of (1)

$x_{1i}^{i,j} + x_{bl}^{i,j} = x_{1il}^{i,j} + x_{b2}^{i,j},\ i=\overline{1,k},\ j=\overline{1,k}$.

We obtain:

- when $i' \neq i$ or $j' \neq j$: 0+0=0+0 and further 0=0;
- when $i'=i$ and $j'=j$: 1+0=1+0 and further 1=1.

In the same way all the 16x7 combinations may be checked. ■

Theorem 1. The net *n2sk* is a p-invariant Petri net for an arbitrary natural number k.

Proof. Let us consider the sum of the sixth and seventh lines of the matrix (2) which represents the solutions of the system (1) according to Lemma 1:

$$\begin{array}{l}
(((p_{1i}^{i,j}, p_{1o}^{i,j}, p_{4i}^{i,j}, p_{4o}^{i,j}, p_{b1}^{i,j}, p_{b2}^{i,j}, p_{b3}^{i,j}, p_{b4}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k}),\\
\qquad ((p_{4i}^{i,k+1}, p_{4o}^{i,k+1}),\ i=\overline{1,k}),\ ((p_{1i}^{k+1,j}, p_{1o}^{k+1,j}),\ j=\overline{1,k}))
\end{array}$$

plus

$$\begin{array}{l}
(((p_{1il}^{i,j}, p_{1ol}^{i,j}, p_{4il}^{i,j}, p_{4ol}^{i,j}, p_{bl}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k}),\\
\qquad ((p_{4il}^{i,k+1}, p_{4ol}^{i,k+1}),\ i=\overline{1,k}),\ ((p_{1il}^{k+1,j}, p_{1ol}^{k+1,j}),\ j=\overline{1,k}))
\end{array}$$

equals to

$$\begin{array}{l}
(((p_{1i}^{i,j}, p_{1il}^{i,j}, p_{1o}^{i,j}, p_{1ol}^{i,j}, p_{4i}^{i,j}, p_{4il}^{i,j}, p_{4o}^{i,j}, p_{4ol}^{i,j}, p_{b1}^{i,j}, p_{b2}^{i,j},\\
p_{b3}^{i,j}, p_{b4}^{i,j}, p_{bl}^{i,j}),\ i=\overline{1,k},\ j=\overline{1,k}),\\
((p_{1i}^{i,k+1}, p_{1il}^{i,k+1}, p_{1o}^{i,k+1}, p_{1ol}^{i,k+1}),\ i=\overline{1,k}),\\
((p_{4i}^{k+1,j}, p_{4il}^{k+1,j}, p_{4o}^{k+1,j}, p_{4ol}^{k+1,j}),\ j=\overline{1,k})).
\end{array} \quad (3)$$

As all the $N_k^{n2p} = 13 \cdot k^2 + 8 \cdot k$ places are mentioned in this invariant, the Petri net *n2sk* is a p-invariant net for an arbitrary natural number k. Moreover, as each component of (3) equals to the unit, the net *n2sk* is a safe and bounded Petri net for an arbitrary natural number k. ■

As the p-invariance was proven for an arbitrary natural number k we say that the invariants of infinite Petri nets with the regular structure were studied.

The proof that (2) is a basis of the system (1) solutions is appreciated. But this fact was only grounded by the calculation experiments for the sequence $k = 1,..,10$. Solutions given by (2) were compared with the basis obtained via the Adriana software (Zaitsev 2006) for the matrix structure with definite k.

T-INVARIANTS AND DEADLOCKS STRUCTURE

For the calculation of t-invariants the same approach may be applied. It gains that the Petri net *n2sk* is not t-invariant, but it is not a surprise because the modeled system is open as the terminal devices are not attached. The simplest way to prove it, is the consideration of the border places without

the input arcs (places without the output arcs as well). So let us consider the net *nf2sk* which is obtained of the net *n2sk* by attaching the terminal devices *nf* represented in Fig. 2b.

But due to the explosion of the basis even for a small enough $k = 3$, the general parametric solution was not constructed. This fact may be easily grounded by the consideration of all the consistent firing sequences of transitions. At first we prove that the net *nf2sk* is t-invariant. For this purpose the consistent firing sequence that contains all the transitions is constructed. We create a transmission graph of the communication matrix. The graph is composed of the cells corresponding to the devices. The cells have the form shown in Fig. 4.

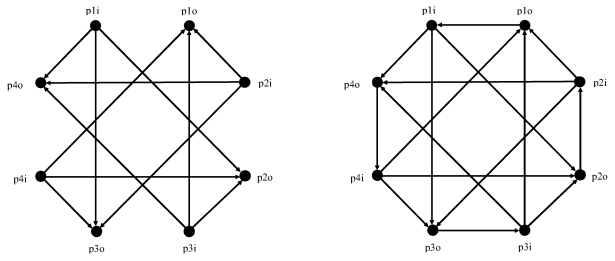
Each arc of the graph corresponds to firing a pair of transitions supplying the movement of a packet to the corresponding port. For instance, the arc $(p1i, p4o)$ represents the sequence $t1i4, t4o$ and so on. For the graph shown in Fig. 4b the following loop may be constructed, which contains all the arcs:

p1i, p2o, p2i, p4o, p4i, p3o, p3i, p1o, p1i, p4o, p4i, p2o, p2i, p3o, p3i, p4o, p4i, p1o, p1i, p3o, p3i, p2o, p2i, p1o

This loop corresponds to the following firing sequence of transitions

t1i2, t2o, tfi-r, tfo-r, t2i4, t4o, tfi-l, tfo-l, t4i3, t3o, tfi-d, tfo-d, t3i1, t1o, tfi-u, tfo-u, t1i4, t4o, tfi-l, tfo-l,
t4i2, t2o, tfi-r, tfo-r, t2i3, t3o, tfi-d, tfo-d, t3i4, t4o, tfi-l, tfo-l ,t4i1, t1o, tfi-u, tfo-u, t1i3, t3o, tfi-d, tfo-d,
t3i2, t2o, tfi-r, tfo-r, t2i1, t1o, tfi-u, tfo-u,

which contains each transition at least once. So the net *nf2s1* is a t-invariant and, moreover, consistent Petri net.



a) without terminal devices (*n2s1*) b) with attached terminal devices (*nf2s1*)

Figures 4: The transmission graph of a communication device

The cells are gathered into the matrix and supplied with the arcs which correspond to the actions of terminal devices for the net *nf2sk*. An example of the graph for $k = 2$ is represented in Fig. 5a.

Theorem 2. The net *nf2sk* is a t-invariant Petri net for an arbitrary natural number k.

Proof. We prove the theorem in a constructive way using the structure of the transmission graph for the net *nf2sk*. We construct the consistent firing sequence that contains all the transitions of the net on the base of the loop of the transmission graph which contains all its arcs. Let us construct the main loop as the composition of loops on the following directions: horizontal, vertical, primary diagonal, collateral diagonal:

1) horizontal loops:

$$((p_{4i}^{i,j} \to p_{4i}^{i,j+1},\ j=\overline{1,k}),\ p_{4i}^{i,k+1} \to p_{4o}^{i,k+1},$$
$$(p_{4o}^{i,j+1} \to p_{4o}^{i,j},\ j=\overline{k,1}),\ p_{4o}^{i,1} \to p_{4i}^{i,1}),\ i=\overline{1,k};$$

2) vertical loops:

$$((p_{1i}^{i,j} \to p_{1i}^{i+1,j},\ i=\overline{1,k}),\ p_{1i}^{k+1,j} \to p_{1o}^{k+1,j},$$
$$(p_{1o}^{i+1,j} \to p_{1o}^{i,j},\ i=\overline{k,1}),\ p_{1o}^{1,j} \to p_{1i}^{1,j}),\ j=\overline{1,k};$$

3) primary diagonal loops:

3.1) left-bottom triangle:

$$((p_{4i}^{v+u-1,u} \to p_{1i}^{v+u,u}, p_{1i}^{v+u,u} \to p_{4i}^{v+u,u+1},\ u=\overline{1,k-v}),$$
$$p_{4i}^{k,k-v+1} \to p_{1i}^{k+1,k-v+1},\ p_{1i}^{k+1,k-v+1} \to p_{1o}^{k+1,k-v+1},$$
$$p_{1o}^{k+1,k-v+1} \to p_{4o}^{k,k-v+1},$$
$$(p_{4o}^{v+u,u+1} \to p_{1o}^{v+u,u}, p_{1o}^{v+u,u} \to p_{4o}^{v+u-1,u},\ u=\overline{k-v,1}),\ p_{4o}^{v,1} \to p_{4i}^{v,1}),$$
$$v=\overline{1,k};$$

3.2) right-upper triangle:

$$((p_{1i}^{u,v+u-1} \to p_{4i}^{u,v+u}, p_{4i}^{u,v+u} \to p_{1i}^{u+1,v+u},\ u=\overline{1,k-v,}),$$
$$p_{1i}^{k-v+1,k} \to p_{4i}^{k-v+1,k+1},\ p_{4i}^{k-v+1,k+1} \to p_{4o}^{k-v+1,k+1},$$
$$p_{4o}^{k-v+1,k+1} \to p_{1o}^{k-v+1,k},$$
$$(p_{1o}^{u+1,v+u} \to p_{4o}^{u,v+u}, p_{4o}^{u,v+u} \to p_{1o}^{u,v+u-1},\ u=\overline{k-v,1}),\ p_{1o}^{1,v} \to p_{1i}^{1,v}),$$
$$v=\overline{1,k};$$

4) collateral diagonal loops:

4.1) left-upper triangle:

$$((p_{4i}^{v-u+1,u} \to p_{1i}^{v-u+1,u}, p_{1i}^{v-u+1,u} \to p_{4i}^{v-u,u+1},\ u=\overline{1,v-1}),$$
$$p_{4i}^{1,v} \to p_{1o}^{1,v},\ p_{1o}^{1,v} \to p_{1i}^{1,v},\ p_{1i}^{1,v} \to p_{4o}^{1,v},$$
$$(p_{4o}^{v-u,u+1} \to p_{1o}^{v-u+1,u}, p_{1o}^{v-u+1,u} \to p_{4o}^{v-u+1,u},\ u=\overline{v-1,1}),$$
$$p_{4o}^{v,1} \to p_{4i}^{v,1}),\ v=\overline{1,k};$$

4.2) right-bottom triangle:

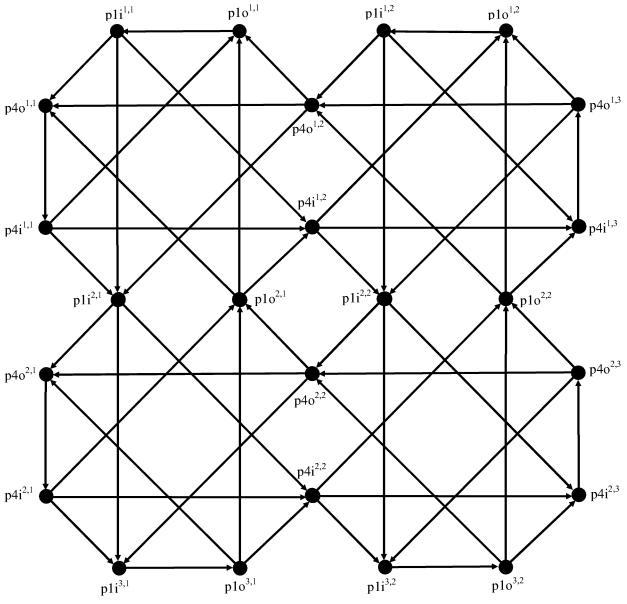
$$(\begin{matrix}(p_{1o}^{k-u+2,v+u-1} \to p_{4i}^{k-u+1,v+u},\\ p_{4i}^{k-u+1,v+u} \to p_{1o}^{k-u+1,v+u},\ u=\overline{1,k-v})\end{matrix},\ p_{1o}^{v+1,k} \to p_{4i}^{v,k+1},$$
$$p_{4i}^{v,k+1} \to p_{4o}^{v,k+1},\ p_{4o}^{v,k+1} \to p_{1i}^{v+1,k},$$
$$(p_{1i}^{k-u+1,v+u} \to p_{4o}^{k-u+1,v+u},$$
$$\begin{matrix}p_{4o}^{k-u+1,v+u} \to p_{1i}^{k-u+2,v+u-1},\ u=\overline{k-v,1})\end{matrix},\ p_{1i}^{k+1,v} \to p_{1o}^{k+1,v}),$$
$$v=\overline{1,k}.$$

On the described loops, firing sequences of transitions may be unambiguously constructed. For instance, the loops for the right-bottom triangle have the following form:

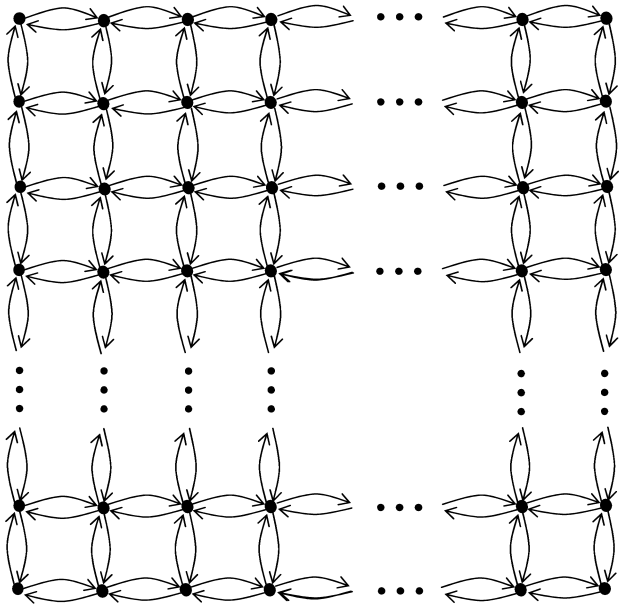
$$((t_{3i2}^{k-u+1,v+u-1}, t_{2o}^{k-u+1,v+u-1}, t_{4i1}^{k-u,v+u}, t_{1o}^{k-u,v+u},\ u=\overline{1,k-v}),$$
$$t_{3i2}^{v,k}, t_{2o}^{v,k},\ t_{fi-r}^{v},\ t_{fo-r}^{v}, t_{2i3}^{v,k}, t_{3o}^{v,k},$$

$(t_{1i4}^{k-u,v+u}, t_{4o}^{k-u,v+u}, t_{2i3}^{k-u+1,v+u-1}, t_{3o}^{k-u+1,v+u-1},\ u=\overline{k-v,1})$,
$t_{fi-d}^{v}, t_{fo-d}^{v}),\ v=\overline{1,k}$.

It is easy to check that the sum of all the firing sequences corresponding to the described loops contains each transition of the net *nf2sk* at least once and preserves the initial marking. So the net *nf2sk* is a t-invariant and, moreover, consistent Petri net for an arbitrary natural number k . ■



a) The transmission graph



b) The graph of possible blockings

Figures 5: Auxiliary graphs (examples for the net *nf2sk*)

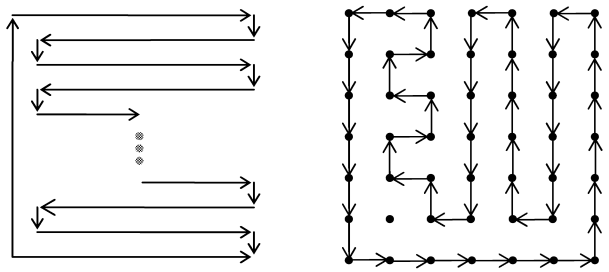
In spite of the fact that the Petri net *nf2sk* is t-invariant and provides the transmission of packets among each pair of terminal devices with redundancy, it contains deadlocks. Deadlocks may occur in the pairs of communication devices but we are more interested in complex deadlocks involving an arbitrary number of communication devices.

Each pair of neighbor communication devices may fall into a local deadlock, for instance, when the device $R^{i,j}$ got l packets directed to the device $R^{i,j+1}$ and the device $R^{i,j+1}$ got l packets directed to the device $R^{i,j}$ and, moreover, the input and output buffers of their common port are occupied with the packets. Such a situation constitutes a t-dead marking for the transitions of both devices while other transitions of the net *nf2sk* are potentially live.

In Fig. 3 the full deadlock for the net *nf2s2* is shown. It involves all the four communication devices of the matrix.

For the description of the deadlocks structures of the net *nf2sk*, the graph of possible blockings shown in Fig. 5b is constructed.

The directed loops of the graph (Fig. 5b) correspond to the deadlocks of the communication matrix *nf2sk*. Each arc connecting a pair of neighbor devices $R^{i,j}$, $R^{i',j'}$, $|i-i'|=1 \vee |j-j'|=1$ means that $R^{i,j}$ may block itself iff it got l packets directed to $R^{i',j'}$, its output buffer of the port connecting $R^{i,j}$ with $R^{i',j'}$ contains a packet and the device $R^{i',j'}$ is also blocked. We may construct a simple chain of arcs and the real deadlock occurs when it is closed in a loop. So deadlocks of the communication matrix may be described as loops of the graph of possible deadlocks. A full deadlock involving all the devices (and all the transitions) occurs when the loop contains all the devices in the matrix. Let us notice that it requires at least $(l+1)\cdot k^2$ packets which should be provided by the terminal devices. Such a deadlock may be easily constructed for an even k using, for instance, the detours of the graph shown in Fig. 6a.



a) For an even k b) For odd $k=7$

Figures 6: The detours of the graphs of possible blockings

For an odd k, the loop may contain only k^2-1 devices but in this case we could make one device isolated by the loop that yields to the full deadlock. So the structure of the

deadlocks is more complicated because, besides the deadlock caused by a cycle of blockings, isolated communication devices may occur with all the four neighbors belonging to the cycle. This case is rather simple for $k = 3$ and illustrated with a full deadlock instance for $k = 7$ shown in Fig. 6b.

In spite of the fact that rather sophisticated square communication matrices were studied, the described deadlocks in the cycles of blockings and isolations are hard-nosed for real-life communication graphs where devices with the compulsory buffering are used. We believe that these deadlocks may be purposely inflicted by the specially situated generators of the peculiar traffic. In real-life networks, the blocking of the devices is overcome with the time-out mechanisms causing the cleaning of the buffers but it leads to a considerable fall of network performance as soon as the situation is repeated by the special generators of perilous traffic.

CONCLUSIONS

Thus, in the present paper, the technique of the linear invariants calculation for infinite Petri nets with the regular structure was presented. The technique was studied on the example of a communication matrix of an arbitrary size but it seems that the obtained results might be generalized for an arbitrary structure as well.

The application of the technique allowed the verification of the telecommunication protocols, involving an arbitrary number of communicating devices. The modeled telecommunication device constitutes a generalized router/switch with the compulsory buffering of the packets. Such positive properties of the communication structure as safeness and consistency were obtained using the linear invariants of infinite Petri nets.

It was proven that the compulsory buffering of the packets inevitably leads to possible blockings of communicating devices. The structure of the complex deadlocks involving an arbitrary number of communicating devices caused by both the cycle of blockings and the isolation was studied.

Though in real-life networks the deadlocks are overcome by the cleaning of the buffers via the time-out mechanism, it leads to a considerable decrease of the network performance and moreover might be inflicted by the ill-intentioned traffic.

REFERENCES

Berthelot, G. and Terrat, R. 1982. "Petri Nets Theory for the Correctness of Protocols". *IEEE Trans. on Communications* 30, No. 12, 2497-2505.

Berthomieu, B.; Ribet, O.-P. and Vernadat, F. 2004. "The tool TINA - construction of abstract state space for Petri nets and Time Petri nets". *International Journal of Production Research* 42, No. 4, (http://www.laas.fr/tina).

Diaz, M. 1982. "Modelling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Model". *Computer Networks,* No 6, 419-441.

Marsan, A.M.; Chiola, G. and Fumagalli, A. 1987. "An Accurate Performance Model of CSMA/CD Bus LAN". *Advances in Petri Nets, LNCS,* Vol. 266, 146-161.

Murata, T. 1989. "Petri Nets: Properties, Analysis and Applications". *Proceedings of the IEEE* 77, (April), 541-580.

Russell, T. 2004. *Telecommunications Protocols*, 2nd Edition, McGraw-Hill.

Shmeleva, T.R. 2007. "Verification of switched Ethernet protocols using infinite Petri nets". In *Proceedings of Third International Conference on Modern Information-Communication Technologies*, (September 24-28, Levadia, Ukraine), 161-163.

Zaitsev, D.A. 2005. *Functional Petri Nets*. Universite Paris-Dauphine, Cahier du Lamsade 224, (www.lamsade.dauphine.fr/cahiers.html).

Zaitsev D.A. 2006. "On realization of compositional algorithms of linear systems solution". *Upravliayuschie systemi i mashini (Control systems and machines)*. No. 3, 32-41. In Russ.

Zaitsev, D.A. and Zaitsev, I.D. 2006. "Verification of Ethernet protocols via parametric composition of Petri net". In *Proceedings of 12th Symposium Information Control Problems in Manufacturing*, (May 17-19, Saint-Etienne, France), 261-267.

Zaitsev, D.A. and Shmeleva, T.R. 2006. "Switched Ethernet Response Time Evaluation via Colored Petri Net Model". In *Proceedings of International Middle Eastern Multiconference on Simulation and Modelling*, (August 28-30, Alexandria, Egypt), 68-77.

AUTHORS BIOGRAPHY

TATIANA SHMELEVA was born in Nizhniy Novgorod, Russia and went to Moscow Institute of Transport Engineers where in 1991 she obtained Ms.D. in Applied Mathematics. She worked in Railway Control Systems Design Centre, Baribino then in Odessa Post Office and Odessa Railway. In 2005 she went to Odessa National Academy of Telecommunications where obtained Ph.D. in Telecommunications in 2008 and where she has been working as a lecturer ever since.

DMITRY ZAITSEV (www.geocities.com/zsoftua) was born in the town Konstantinovka, Donetsk region of Ukraine. He got Ms.D. in Applied Mathematics from Donetsk National Technical University in 1986, Ph.D. in Automated Control from Kiev Cybernetics Institute in 1991, Dr.Sci. in Telecommunications from Odessa National Academy of Telecommunications in 2006. At the present time he works as a Professor of Communication Networks Department in Odessa National Academy of Telecommunications (www.onat.edu.ua).

IVAN ZAITSEV was born in Donetsk, Ukraine and went to Novosibirsk State University, Russia where in 2008 he obtained B.D. in Information Technology. At the present time he is a master student.

MODELING AND SIMULATION OF AN INTRUSION TOLERANT WEB SERVICE ARCHITECTURE USING COLOURED PETRI NETS

Zahra Aghajani
Mohammad Abdollahi Azgomi
Performance and Dependability Engineering Lab., School of Computer Engineering
Iran University of Science and Technology, Tehran, Iran
E-mail: z_aghajani@vu.iust.ac.ir and azgomi@iust.ac.ir

KEYWORDS
Modeling and Simulation, Intrusion Tolerant Web Service, Coloured Petri Nets.

ABSTRACT

An intrusion is a malicious operational fault resulting from successful attacks on system vulnerabilities. Intrusion tolerance is a fault-tolerant design approach to defend information systems against malicious attacks. Abandoning the conventional aim of preventing all intrusions, intrusion tolerance instead relies on triggering mechanisms that prevent intrusions from spreading on the system and leading to security failures. ITWS in a multi-layer architecture for intrusion tolerant web services, which is based on single version fault tolerant techniques. In this paper, we present a coloured Petri net (CP-net) model for describing the details of the ITWS components. In addition, we have done some simulations to evaluate some performance measures of the architecture.

INTRODUCTION

Network attacks are treated as maliciously devised conditions to exploit design, implementation or configuration faults and intrusions are treated as security failures (Reynolds et al. 2001). Intrusion tolerance is the ability of a system to continue providing adequate services after penetrations. This means that the system is designed so that the damage caused by the intruder is contained and perhaps automatically repaired (Stavridou et al. 2001).
A web service is any service that is available over the Internet, uses a standardized XML messaging system, and is not tied to any operating system or programming language. Interoperability, self-containing and self-describing are the main features of these services, which make them suitable to implement complex and distributed service-oriented applications over the Internet (Cerami 2002). Service-oriented architecture can help to prevent companies to rebuild existing applications, improving the reusability and maintenance cost. However, these architectures are often unstable due to the unreliability of web services on movement, deletion and subjecting to various sources of failures and attacks. Therefore, the ability to withstand intrusions and faults becomes highly important when considering the availability and survivability of these services.
In this paper, we present a coloured Petri net (Jensen 1992, Jensen 1994) model for modeling the ITWS architecture by using CPN Tools (CPN Group). We use the token game animation feature of this tool to check the correctness of models and have done some simulations to evaluate some performance measures of ITWS. The rest of this paper is organized as follows. Section 2 introduces the ITWS. In section 3, a CP-net model of the architecture is presented. Section 4 presents the simulation results of the architecture. Finally, section 5 concludes the paper.

AN OVERVIEW OF THE ITWS ARCHITECTURE

The ITWS is a multi-layer architecture that tries to use single version fault tolerant techniques to tolerate malicious failure. Fault tolerant techniques are designed to work against faults that are modeled as rare events occurring independently, but malicious faults and security attacks on the other hand are intentional, systematic, and repeatable. Therefore, standard fault tolerance techniques cannot be directly applied to obtain intrusion tolerance and new solutions are needed (Stavridou et al. 2001).
Four fundamental building blocks in ITWS form different layers of security in key parts of the system to detect, prevent, confine, and tolerate attacks. This architecture uses the simple primary-backup scheme. These two services have the same implementations and therefore they will have same vulnerabilities. However, if we could prevent previously detected attacks from occurring continually on the system, we can use the benefits of single-version fault tolerance techniques even in case of malicious failures.
The web service proxy-firewall forms the preventive part of this architecture. Since malicious SOAP messages can potentially cause harm to critical computing assets, this application level gateway performs some grammatical validations on SOAP messages to discard malicious and suspicious requests/responses. It uses an attack pattern database, filtering policies and valid SOAP requests/ responses schemes to perform validations.
The host-based IDS module follows the behavior of elements in the application level to detect any abnormal behaviors. It uses host-based sensors and log files to detect malicious sequences. By detecting any new attack, it updates the attack pattern database to help the web service firewall to reject this kind of malicious requests. It also sends appropriate alerts to the reconfiguration module, which is responsible to handle detected instabilities and returns the system to a healthy state.
The containment module confines the effects of successful attacks to a subset of the system. It uses two fundamental components: a vulnerability dependency matrix and an intrusion graph. The vulnerability dependency matrix is a comprehensive model that defines the effect of all known and potential attacks to the system and is set by the system

administrator. By announcing any new vulnerability, this matrix updates and specifies the effects of the newly detected vulnerability on system services. An intrusion graph is a directed graph shows the path for intrusions to spread from one service to its neighbors. The containment module receives the alerts from network-based intrusion detection systems and detects the compromised services. By detecting any compromisation, it sends an alert to the reconfiguration module to disable that service. In this definition, the system services contain both application level services and system services.
The reconfiguration and recovery module is the main part of the ITWS. It responds to detection and containment alerts, detects faulty services by sending test requests and checks the level of services on the primary, replaces the backup if necessary. It also protects the application environment unauthorized accesses.
For more information about ITWS, please see (Aghajani and Abdollahi 2008)

A CP-NET MODEL OF THE ITWS ARCHITECTURE

In this section, we present a CP-net model of the architecture. To reduce the complexity and make the model more readable, we use hierarchical structures. In these models, we model complex subsystems as substitution transitions. Figure 1 shows the top-level view of our model. The *Request* place is the start point of the model and contains the list of input requests. The *WSF* substitution transition models the WS firewall, validates each input requests and inserts accepted requests to the *LogFile* place.

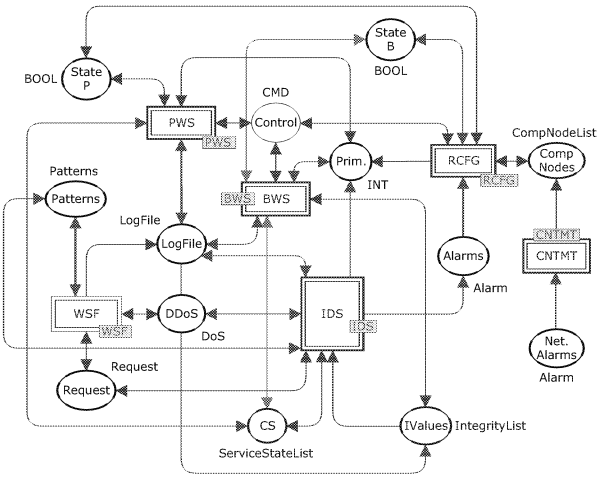


Figure 1: Top-Level View of the CP-net Model

The *DDoS* place is a logical place which is updated by the *IDS* transition and specifies the Denial of Service attack conditions. If the system enters to the DDoS attack conditions (by flooding the service), the WS firewall module will authenticates the sender of each request to detect and reject requests from spoofed IP addresses. The *IDS* transition monitors the state of the active services and critical resources on active primary periodically and sends appropriate alerts to the reconfiguration module.
The *IValues* (Integrity Values) and *CS* (Current Services) places are the input ports for the *IDS* module and represent information about the active primary environment. The *stateP* and *stateB* places have Boolean colors and determine whether the active backup system is ready to be replaced by the active primary, if it is in a failed or degraded state. The *Prim* place shows the current active primary. The *CNTMT* transition models the containment module; sends appropriate alarms to the CNTMT manager to disable compromised services.

Model of Web Service Firewall

Figure 2 illustrates the submodel of the *WSF* transition in the main model. In this mode, each request is specified by a vector of (*Service URI, Http Header, SOAP Message*) and is sent to the *Get HTTP Request* transition. If the system is under any DDoS attack, it uses the *Redirection()* function and returns the request back to the start state. The purpose of this function is to detect malicious traffics from legitimate requests. If the attacker uses spoofed IP addresses, he/she could not be able to receive the redirection messages to resend the request to the new location. We use a Bernoulli distribution function to model if a request has a legitimate or spoofed sender. If it returns one, it means that the requester is alive, receives the redirection message and resends the request. Otherwise, it is a fake request that may be used to flood the service and must be rejected.
In DoS attack conditions and for each received request, the web service firewall checks the header part of the requests. Newly arrived requests, are redirected by the *Redirection()* functions and previously redirected requests are forwarded to the *ATHENT*. The *VI()* function validates the redirected messages with their redirection identifiers. If a request has an invalid identifier, the *ATHENT* set the *Next* state to true to let the next request to enter to the system. In this model, the web service firewall validates each request in four different steps: maximum request length, valid characters, known attacks and valid SOAP schema.
Successful validation in each step passes the request to the next step. If a message passes all the four steps successfully, the WS firewall inserts it to the *LogFile* place; otherwise, it will be discarded and the next request will be entered to the system. The *LogFile* stores data in an ordered pairs of *(Request, Response)*. In DoS attack conditions, the web service firewall redirects only legitimate requests. From the viewpoint of the web service firewall, a request is a legitimate request if it has the value *one* in its header part. After the redirection phase, the value of this header is changed to the length of the SOAP message which is considered to be greater than one and the *ATHENT* performs ID validations on requests which their header values are greater than *one*.

Intrusion Detection Model

Figure 3 shows the submodel of the *IDS* substitution transition. As we mentioned earlier, this module is a host-based intrusion detection and response system tries to detect abnormal behaviors or misuses in the service environment. It monitors the status of the system services, file accesses, checks the integrity requirements of the operational environment and sends appropriate alerts to the reconfiguration system. It also detects if the system is under

DoS attack conditions. To reduce the complexity of the model, we implement the *DoS* and *Unauthorized modification* as substitution transitions.
The two *Alarm* and *DDoS* places are output port of this component. Each alarm is represented by the (*active primary, operation, parameters*) vector. The active primary filed specifies which of the primary or backup is operational now. We should mention that in this model, there is an implicit ID initialized by *zero*. We use this ID to introduce the active primary to other parts of the system. An even ID represent that the main primary is the active primary and an odd ID represents that the main backup is the active primary now. After each replacement, this ID is incremented by one to help us to respond to alerts that are generated for the current active primary. The reconfiguration module uses the two other fields to respond to the detection alerts.
The *current services* port feeds the detection system by the name and status of system services that are running on the active primary. The reconfiguration module updates these two lists periodically. Furthermore, the system administrator initiates *required services,* which are necessary for the normal operations of the system, too. This information is represented in (*service name, status*) pairs. By comparing these two lists in the *Service State* transition, the IDS module becomes enabled to detect unauthorized changes in the state of services. If there is an illegal active service in the list of current services the intrusion detection system sends an (*active primary, "Disable", service name*) alert to the reconfiguration module; otherwise, if there is a service in the required services list, which is not active now, it sends an (*active primary, "enable", service name*) alarm, respectively. These alarms are placed in the *Alarm* place, an intermediate place between these two subsystems.
We should notice that the *"Service state"* transition compares the head element of these two lists by each other in each step and repeats it until one of the lists becomes empty. Therefore, if the length of these two lists is not equal, the elements of the longer list will not be processed by this transition according to the execution rules of Petri nets. The *"Make Compat."* transition makes the length of these two lists equal by inserting enough empty pairs to the smaller list. Similarly, the *Invalid File Access* transition audits the access log file to detect if a process has unauthorized accesses to unauthorized resources. Tokens in the *Access Log* and *Valid Access List* places are stored in (*Service name, Access resource*) pairs. The *InvalidAccess()* function is responsible to send misuse detection alerts.

Check Integrity of the Operational Environment

The ITWS uses file checking techniques to detect unauthorized modifications on the system resources. This technique uses cryptographic checksums on sensitive system resources to detect changes including unauthorized software installations, back doors left by successful intrusions and system corruptions (Reynolds et al. 2001).
In this model, we assume that the system administrator specifies critical directories and files to the integrity checker module. This module computes the hash values of each file once the system is in a healthy state. It then compares these hash values with which ones it calculates periodically to detect if any unauthorized modifications, deletions or creations is happened. Figure 4 shows how the system detects unauthorized modifications in our architecture.
The place "*Hash Values*" contains the list of file names and their hash values in a healthy state. Similarly, the place *CIValues* represents the list of current integrity values and is set by the *WSP* or *WSB* subsystems respectively. The two *Decomp* and *Decomp1* transitions use the *Decomp()* function to divide the list of integrity values which are stored in (*File name, Hash value*) lists to two separate lists: one for file names and the other for relevant hash values. Therefore, the places *FName* and *FName1* will contain the name of the main files and sample files names and similarly the two places *HValue* and *HValue1* represents their hash values stored in the list data type.
Comparison is done in two levels, first to detect unauthorized creations and deletions and second to detect unauthorized modifications. According to the fact that hash functions generate unique hash values, any difference between hash values of a data file in different intervals means that the content of the file or some of its attributes are changed. After decomposing integrity values, the *"Unauthorized Modification"* transition starts to check the integrity of system environment.
The recursive function *CD()* is used to detect unauthorized creations and deletions. It compares the main files names and sample file names to detect creation or deletions. It does not need to check their hash values in this step. If there is any file name in the main files name list that is not in the sample file names list, it generates an (*Active primary, "Create", "File name"*) alarm in the *Alarm* place. Similarly, if there is a file name in the sample list which is not in the main list, an (*Active primary, "Delete", "File name"*) alarm will be generated.
Places *FN* and *HV* are as same as the *FName* and *HValues* places. Every time the *"Unauthorized Modification"* transition fires, it removes the first element from the *FName* and *HValue* list. However, since we need this default information in the next steps, for each element that is removed from these two lists, we add its name and its hash value to the *FN* and *HV* places.
Each time the *"Unauthorized Modification"* transition fires, it calls the *HUP()* function. If the head element of the *FName* list exists in the *FName1* list, it uses the *ins_new* function to add the head element of the *HValue* list to the *HValue1* list. Now, if the length of the *HValue1* list becomes longer than the length of the *FNames1*, it detects that there is a critical file that its hash value is different from the desired hash value, meaning that the file has been modified. By transforming these newly inserted hash values to the *"All modified*" place, we use the *indx()* function to find the index of each hash value in the main hash value list and use this index to find its relevant file name in the main file name list and send an alert (*Active primary, "Replace", File name*) to the *Alert* place. Hence, this mechanism help us to combat DoS attacks when an attacker removes critical resources from the target system or detect unauthorized modifications on these resources when he/she tries to change the workflow of the system maliciously.

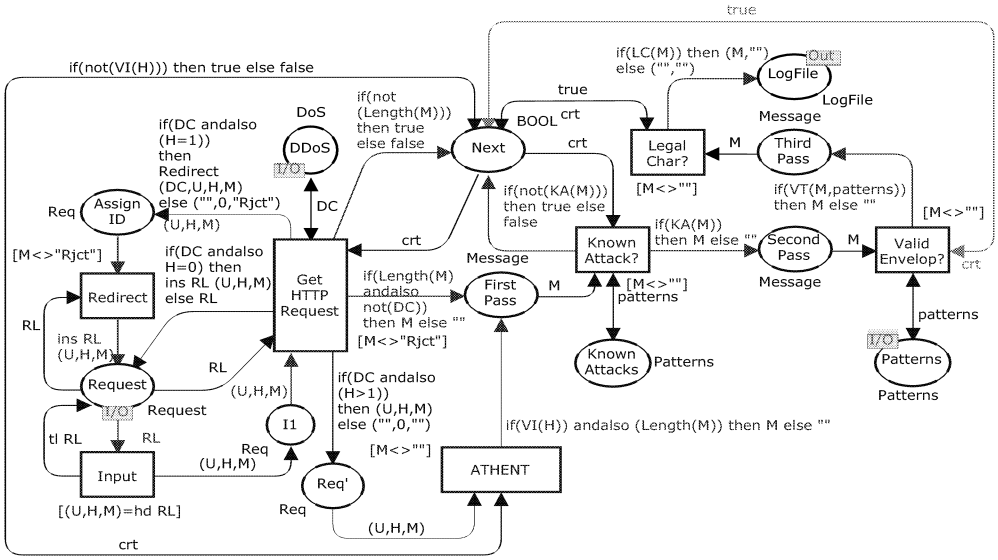


Figure 2: Submodel of the WSF Substitution Transition

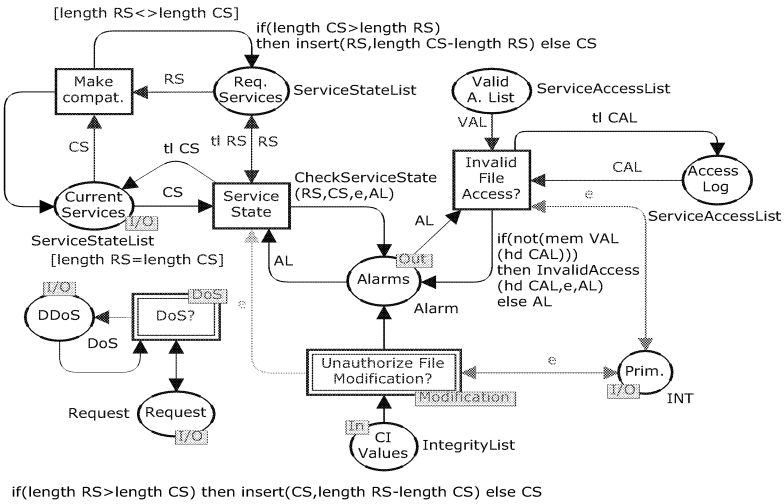


Figure 3: Submodel of the Intrusion Detection System

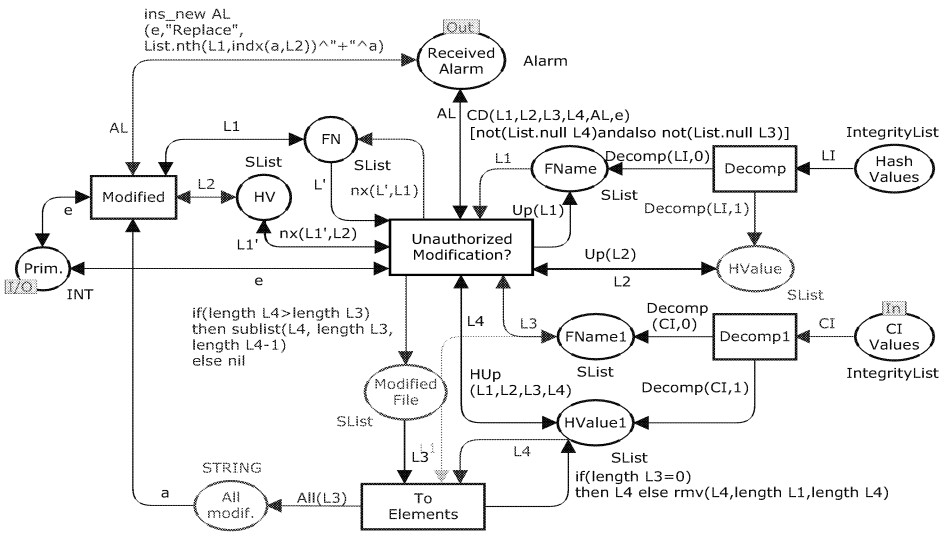


Figure 4: Monitoring the Integrity of the Operational Environment

Figure 5 models the DoS attack detection mechanism. In this model, we assume that if the number of requests becomes more above a threshold, the system enters to the under-attack mode. The input/output port *Request* represents the list of requests waiting to enter to the system. Waiting requests are specified by the value *0* in their header field. To isolate malicious traffics from legitimate traffics, the IDS module counts the number of requests from each sender until the system is in the under-attack mode. The *DoS* transition counts the number of waiting requests by using the *DoS*() function. If the number of requests becomes above a predefined threshold, it sets the *DDoS* place to *true* to force the web service firewall to use its redirection/authentication mechanism to detect requests from spoofed IP addresses. The place *"IP/ID"* maintains a list of IP addresses of senders in attack time intervals and the place *Count* stores the number of requests from each sender. For each newly received request, if its IP address exists in the *IP/ID* list, the *Flooding* transitions increments its relevant count number by *one* by using *Inc()* function. This request will be rejected if the number of requests that are sends from its sender exceeds from an accepted threshold by using the DoSR() function.
Rejected requests are labeled by replacing their message filed with the "*Rjct*" message. The web service firewall will not allow rejected requests to enter to the system. If the receiving request becomes the first request from a sender, the *"Update Request Count List"* transition becomes enabled and adds its IP address to the *IP/ID* list with count number *one* in the count list. Legitimate requests are labeled by setting their header filed with the value *one*. The web service firewall considers these requests as legitimate requests in DoS attack conditions. Each time the system returns to the normal state, the *clear* transition becomes enabled and clears the list of IP addresses and the count list.

The Reconfiguration Module

Figure 6 shows the submodel of the reconfiguration (*RCFG*) transition in the main model. The *Alarm* place is the main input port for this submodel. The reconfiguration module makes appropriate decisions according to the alarms from intrusion detection system or containment module.
The place *Cmd* is an intermediate place between this module and the two primary and backup systems. It sends appropriate commands to the *Cmd* place in (*Active primary, Opcode, Operand*) format. The primary and backup systems respond to the appropriate commands according to the active primary filed. The both *P* and *B* places represent the state of the primary and backup system if they become ready to be replaced by the active primary and are set by the *PWS* and *PWB*, respectively.
The active primary service must report the level of services it provides to the reconfiguration manager periodically. To do this, it uses the opcode *0* and sends (*Active primary, 0, Level*) message to the *Cmd* place.
The *"Set configuration"* transition gets this information to make appropriate decisions about reconfiguring the active system. Each time this transition fires, it moves an alarm from the *Alarms* state to the *Cmd* place and sets the *Next* place to *true*, lets the next alarm to enter to the *Alarm* place.
The place *Pairs* contains sets of pre-defined and critical request/response pairs. The *"Random Req/Response"* transition checks the active primary to detect if it is in a healthy state or not. When this transition fires, it chooses one of these pairs randomly, puts the result part in the *Result* Place and issues a command with opcode *1*. The active primary responds to this command and puts the result on the *Cmd* place with opcode *2*. The *"Faulty Service"* transition compares the desired response with the service response to detect whether the service is healthy or not. If it detects that the service is faulty, it checks the state of the backup system by using *Stdby()* function. If the backup system becomes ready, it will be replaced by the faulty active primary. As we mentioned earlier, an even ID represent that the main primary is the active primary and an odd ID represents that the main backup is the active primary now. Therefore, to do the replacement we only need to increment the identifier of the active primary. After any replacement, previously generated alarms for the last active primary must be cleaned. This is done by *Clear Last Alarms* transition. It uses the *Clear()* function and removes alarm that their identifier is less that the current primary ID.

The Web Service Module

Figure 7 illustrates the submodel of the *PWS* (primary web service) transition in the main model. The *BWS* transition has the same model. The *"Primary Service"* transition is an abstract transition that is modeled according to each service specifications. The *"Service Host"* transition is responsible to report about service environment states and responds to the commands that were sent from the reconfiguration manager. The *LogFile* place is an intermediate port between the two *WFS* and *PWS* substitution transitions. Each time the *Primary Service* fires, it takes a request from the log file and puts its response to the *LogFile* place and sets the *Next* place to *true* to let the next request enters to the system. The *CI* and *CS* places contain the list of current integrity values and current service names. The *"Service Host"* transition uses the *IL()* and *SL()* functions to update these two lists every time it fires. These functions insert or remove randomly some elements from the required services and integrity information lists to model the dynamic behavior of the systems in their working time.
For each command with opcode *3*, the *"Service Host"* transition calls the *RCIL()* and *RCSL()* functions to reconfigure and update the list of current services and current integrity values. This means that, it responds to the IDS alarms by using these two functions. We should only noticed that the *RCSL()* function does not restart services which are disable due to the containment alarms. Because, these services are detected as malicious and are disabled to confine the spread of intrusion.
Commands with opcode *4* mean that the active service must be restarted. Therefore, by receiving this command, the *State* place is set to *false* and the *Count* place is incremented by one. In this model, we assume that the restart operation continues as long as the *Service Host* fires five times. When the count number becomes five, the *State* place is set to *true* and the *CI* and *CS* places are updated by the healthy information. The *DIL()* and *DSL()* functions provides default integrity and default service lists. The *Healthy* transition fires

each time a replacement occurs to return the state of the offline system back to a healthy state. Detecting replacement is done by checking whether the current identifier is even or not.

SIMULATION RESULTS

To evaluate the proposed model, we have simulated the model and have gathered information by using the monitor mechanism of the CPN Tools. For this evaluation, we assume that the whole system may be in one of the *Healthy*, *Degrade* or *Fail* states. The system will be in a *Healthy* state, if at least one of the primary or backup systems is in the healthy state. Failures and repair mechanism are modeled by exponential distribution functions. In the *Healthy* state, the system could be able to answer to all requests. The response mechanism is also modeled by an exponential distribution function.

In *Degrade* mode, requests for the disabled subsystems are rejected and in the *Fail* mode, the system does not answer to received requests. Figure 8 shows the percent of responded requests in total number of requests. As shown in the figure, the effectiveness of this model, especially in large failure rate, depends on how fast the repair mechanisms are done. In Figure 9, we measure the same parameters to check the effectiveness of the containment module. If the system does not confine compromisation, the failure rate will increase since failures and compromisation will be cascaded in the system, causing much more number of requests to be rejected.

CONCLUSIONS

ITWS is a multi-layer architecture for intrusion tolerant web services. Four fundamental building blocks in this architecture form different layers of security in key parts of the system to detect, prevent, confine and tolerate attacks. In this paper, we presented a CP-net model of the ITWS architecture. Each part of the system is modeled separately in a different page that is associated with other pages by using appropriate input/output ports. We use the token game animation feature of the CPN Tools to check the correctness of the overall model. In addition, we have simulated the model to evaluate some performance measures of the architecture.

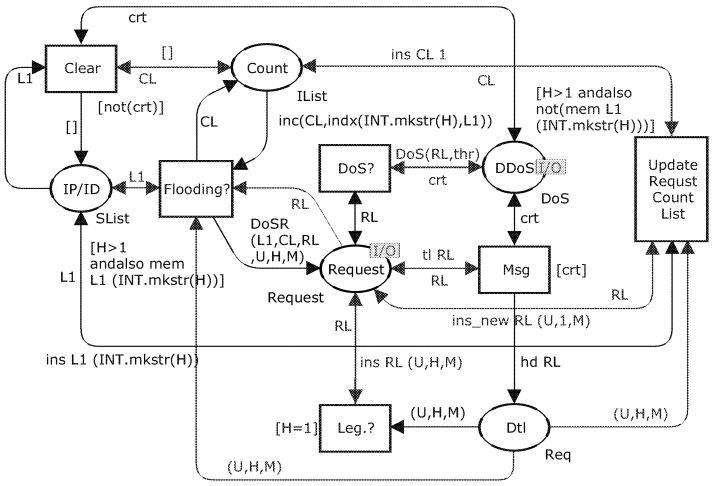


Figure 5: Detecting Denial of Services Attacks

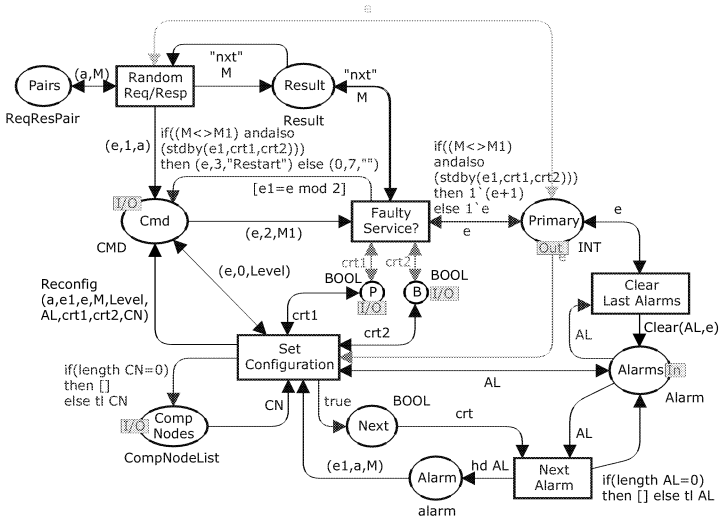


Figure 6: Submodel of the Reconfiguration (RCFG) Transition

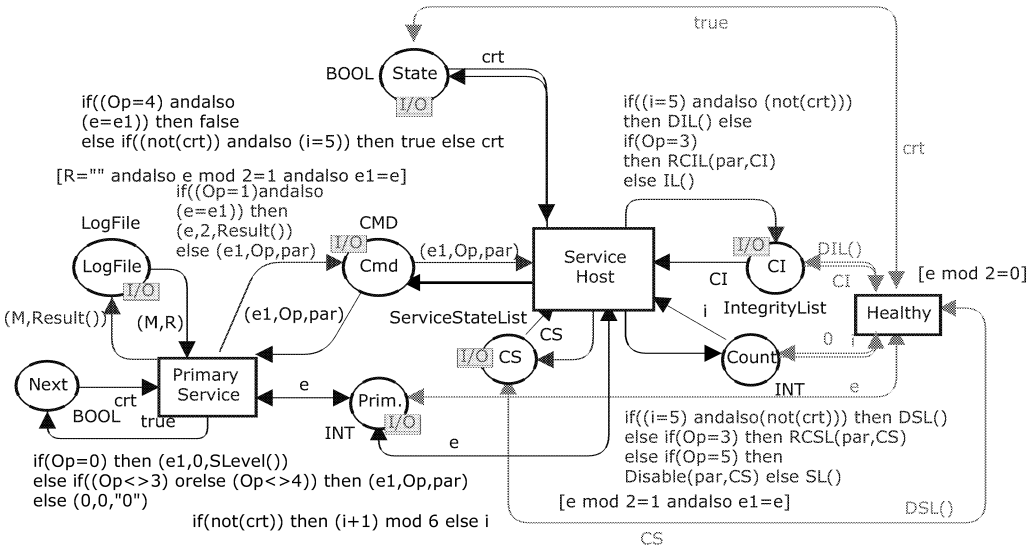


Figure 7: Submodel of the PWS Substitution Transition

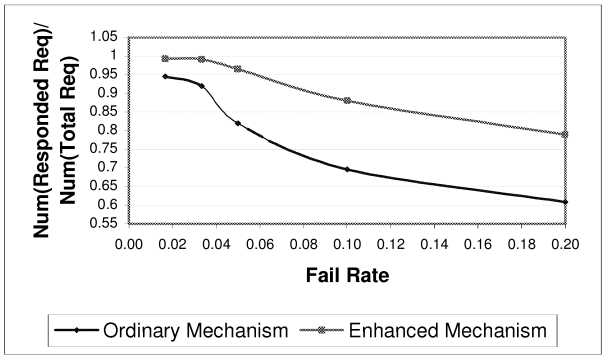


Figure 8: Simulation Results, Improving the Repair Mechanisms

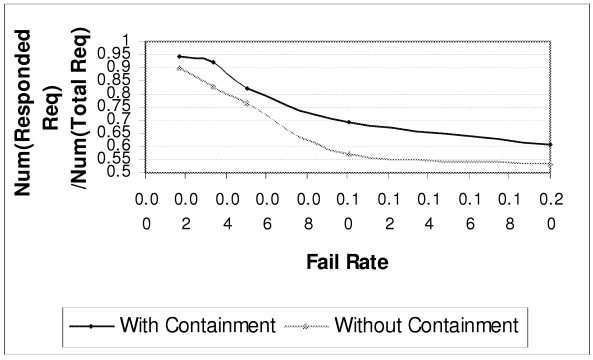


Figure 9: Effectiveness of the Containment Mechanism

REFERENCES

Aghajani, Z. and M. Abdollahi Azgomi. 2008. "A Multi-layer Architecture for Intrusion Tolerant Web Services." *Int'l J. of u- and e-Service, Science and Tech.*, Vol.1, No.1, SERSC, 73-80.

Cerami, E. 2002. *Web Service Essentials.* O'Reilly, 6-24.

CPN Group. "CPN Tools." University of Aarhus, Denmark, URL: *http://wiki.daimi.au.dk/cpntools.*

Jensen, K. 1992: *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Basic Concepts*. EATCS Monographs in Theoretical Computer Science, Springer.

Jensen, K. 1994. "An Introduction to the Theoretical Aspects of Coloured Petri Nets." *LNCS*, No. 803, Springer, 230-272.

Reynolds, J.; et al. 2001. "The Design and Implementation of an Intrusion Tolerant System." In *Proceedings of Int'l Conf. on Dependable Systems and Networks (DSN'01)*, 285-290.

Stavridou, V.; et al. 2001. "Intrusion Tolerant Software Architectures." In *Proceedings of DARPA Information Survivability Conf. and Exposition*, Volume II, 230-241.

Verwoerd, T. and R. Hunt. 2002. "Intrusion Detection Techniques and Approaches." *Computer Comm.*, Vol. 25, 1356-1365.

BIOGRAPHY

ZAHRA AGHAJANI received the B.Sc. degree in computer engineering (software) from Islamic Azad University (2003) and M.Sc. degree in information technology from Iran University of Science and Technology (2009). Title of her M.Sc. thesis was "A Multi-Layer Architecture for Intrusion Tolerant Web Services" which has been done under supervision of Dr. Mohammad Abdollahi Azgomi. Miss Aghajani's research interests include system modeling and web service. She has published several papers in international conferences and journals.

MOHAMMAD ABDOLLAHI AZGOMI received the B.Sc., M.Sc. and Ph.D. degrees in computer engineering (software) (1991, 1996 and 2005, respectively) from Sharif University of Technology, Tehran, Iran. His research interests include performance and dependability modelling with high-level modelling formalisms such as stochastic Petri nets, tools for modelling and evaluation, verification and validation, object-oriented modelling, web services, network and web security. He has published several papers in international journals and conferences. Dr. Abdollahi Azgomi is currently an assistant professor at the school of computer engineering, Iran University of Science and Technology, Tehran, Iran.

A CYBER ATTACK SIMULATION MODEL FOR EVALUATION OF SYSTEM AVAILABILITY

Golriz Khazan
Mohammad Abdollahi Azgomi
Performance and Dependability Engineering Lab., School of Computer Engineering
Iran University of Science and Technology, Tehran, Iran
E-mail: khazan.golriz@gmail.com and azgomi@iust.ac.ir

KEYWORDS
Attack Simulation, Quantitative Security Evaluation, Availability, SimEvents.

ABSTRACT

As a result of the growing attention to the security of computer systems and networks, security evaluation becomes very significant. However, on defection of security assessment methodologies, like ITSEC, CC and etc., the absence of a unified framework for modeling and quantitative security evaluation (QSE) is obvious. In spite of the significance of QSE, a solution based on simulation in this field is not offered yet. In this paper, simulation of a computer system for a QSE based on discrete-event simulation (DES) and SimEvents is presented. First, the system is in normal state and then the arrival of an attacker is simulated and availability of the system, as an important security measure, in any moment of simulation time is evaluated. At the end, a case study on denial of service (DoS) attack simulation in the category of "packet bombardment" is presented and the availability measure is evaluated.

INTRODUCTION

As a result of extremely propagation of Internet and pervasive viruses, worms, Trojans, intruders and electronic spoofing, security has found a very significant role. Computer security event is a state change in a limited computer system from a desirable state to an undesirable state that this state change is issued by an external irritant application to the system (Stephenson 2004).
Security assessment methodologies and standards, such as *information technology security evaluation criteria* (ITSEC) (ITSEC Webpage) and *common criteria* (CC) (CC Webpage) are widely used. However, two important disadvantages of these methods are as follows: (1) the cost of their implementation (2) the time consumed for their accomplishment. Another disadvantage is that these assessment methodologies are not applicable in system design phase.
The efforts used for *quantitative security evaluation* (QSE) of systems are typically based on Markov chains, formal methods and red-teams, but a unified framework for modeling and QSE is an open problem yet.
There are a huge number of academic and commercial simulation languages and tools. Due to several important reasons, we have selected discrete-event simulation (DES) using *SimEvents* (SimEvents Webpage). SimEvents extends *Simulink* with a discrete-event simulation model of computation. With SimEvents, activity-based models of systems are developed to evaluate system parameters, such as congestion, resource contention and processing delays. It is possible to configure entities with user-defined attributes and then aggregate entities and attributes to model data hierarchy and transport in applications such as packet-based networks, mission planning, supervisory control, real-time operating systems and computer architecture (SimEvents Webpage).
In two previous papers (Khazan and Abdollahi 2008 and Khazan and Abdollahi 2009) we presented the proposed framework for attack simulation and quantitative security evaluation using SimEvents and the results of some case studies on attack modeling. In this paper, the simulation of a computer system for quantitative security evaluation using *SimEvents* is presented. For this paper, the system is initially in its normal state. Then, the arrival of an attacker is simulated and the availability of the system, as an important security measure in any moment of simulation time is evaluated. At the end, a case study of denial of service (DoS) attack simulation in category of "packet bombardment" is studied and the availability measure of the system is evaluated.
The rest of this paper is organized as follows. In section 2, some related works are briefly introduced. In section 3, a simulation model for evaluation of system availability is presented. In section 4, as a case study, DoS attack simulation is presented in two variants. Finally, some concluding remarks are mentioned in section 5.

RELATED WORKS

The use of discrete-event simulation for QSE can solve the problems and drawbacks of the existing methods. Till now, the common application of simulation in security has been to use the simulation tools in ordinary networks to model systems and traffics that demonstrate attacks. By security measurements affecting the performance of systems, a useful and obvious way for simulation is acquired; however the suggested security measurements are limited and expensive for the specified applications (Nicol 2005). We can mention some examples of Nicol and his colleagues who have worked directly on border gateway protocol (BGP). The BGP example consists of encryption and decryption takes time and can thus affect the system performance. Additionally the BGP system is so large that we should use simulation to capture the system dynamics (Nicol 2004).
Other related works in this field are as follows. The probability security meter model introduced in (Sahinoglu 2007), receives inputs like vulnerability, threat, lack of countermeasure and constants like criticality, utility cost and

then measures residual risk and cost needs for avoiding from risk.
Simulation-based analysis of security is used in mobile ad-hoc networks (MANET) is another activity in this field in which the impact of network performance threats by means of dynamic resource routing (DSR) is studied (Porcarelli 2004).

A SIMULATION MODEL FOR EVALUATION OF SYSTEM AVAILABILITY

In this section, a simulation model is presented to measure the system availability. In our model two subsystems as hardware-software failures and security failures are used. Hardware-software failures are unintentional, but the security failures are supposed to be intentional. Therefore, the availability of system, as a quantitative security parameter is measured.
In the model shown in Figure 1, the *Entity Time-Based Generator* block generates failure entities with specified exponential distribution, mean=50 (a system with exponential distribution mean=50 fails). When the system detects the security failures, it starts to fix them. The time that the system consumes to repair the security failures can be varied, so, in the model, a *Random Number Event Generator* with uniform distribution between 40 and 70 ($U(40, 70)$ is used to generate random numbers as a repair time. This number is used as the *service time to failure* entity or the *system failure time*. After arrival of the failure entity, the server begins to repair the failures and will not serve other entities. When the lower queuing system generates an entity, changes in its server's #*n* signal invoke the state flow block that determines the state of the upper queuing system. Increasing the #*n* signal, causes the server to downgrade, while decreasing causes the state flow output signal, *Server-Up* becomes one and registers in memory by a latch and then gate enables. Therefore, the server becomes operable again and available.
Output of this model is the system availability ($A(t)$). By increasing the mean value of the failure entity generator, $A(t)$ will decrease. The result is shown in Figure 2.

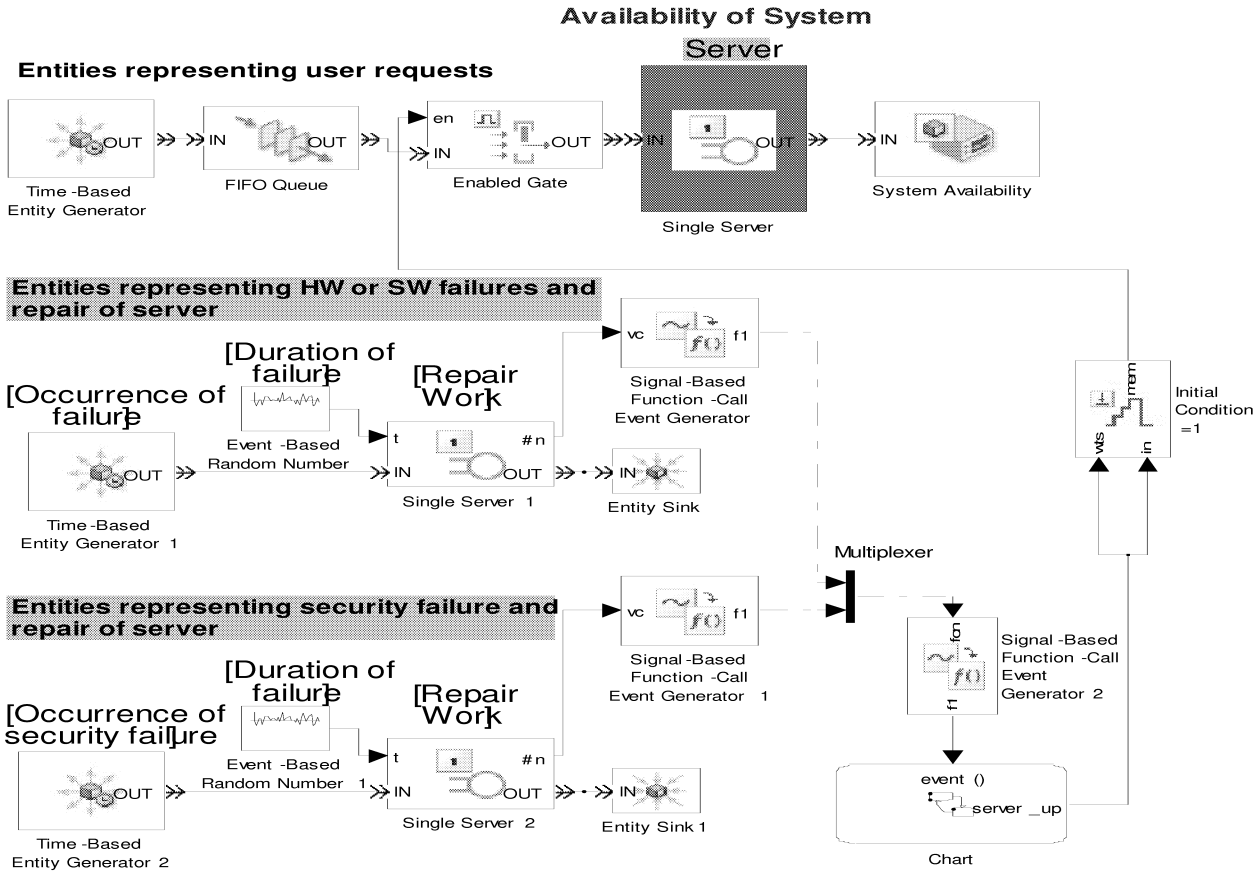


Figure 1: Simulation Model for Evaluating System Availability with Probability of Hardware-Software and Security Failures

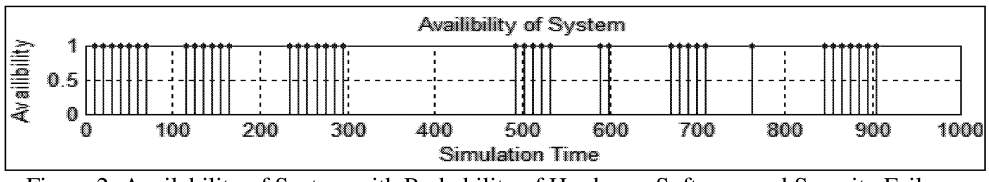


Figure 2: Availability of System with Probability of Hardware-Software and Security Failures

A CASE STUDY: DOS ATTACK SIMULATION

The goal of many attacks is to sabotage and interrupt serving in network systems. In denial of service (DoS) attack, intruder(s) endeavors to stop serving of a server in network by means of a practical/scientific way such that the system is forced to reset or down. These attacks can be started from a local network (locally DoS) or by direct lead of intruder starting from remote (remotely DoS). On technical aspect, DoS attacks fall in two categories:

- Attacks that directly cause server to stop.
- Attacks which cause server overflow and resource destruction.

In this study, the focus is on the second category, which is *locally DoS bombardment*. In this case, attacker tries to send sequential packets to victim to make the server busy, so it cannot reply to real requests and it is bombarded idiomatically (Malekian 2004).

A Simulation Model for the Normal State

The model shown in Figure 3 consists of two clients in left, two routers as source and destination in middle, and three servers in right. The clients generate packets with specified length and send them to the specified target server nominated in its packet. Packets are generated by *Time-Based Entity Generator* with mean=8 and 6 from SimEvents library. Packets length and destinations are generated by *Random Number Event-Based Generator* with uniform distribution. Destination and length can be a number between 1 and 3 and 6 and 10, respectively. Destinations are target servers. To set the specification of each client, *Set Attribute* blocks are used. First attribute, *A1* or *Source* is source packet generator for the first client is set to 1 and for the second one is 2. The second attribute, *A2* or *Length* is the packet length and its value is specified from the *random number generator block* connected to the *set attribute block*. The packets after generating are routed by source router to the destination server that is specified in the *set attribute block* of its own generator. The source router has two ports that each one has a limited capacity queue to hold and route packets. The capacity of both queues is 25 packets and no preemption is defined for passing packets from queues.

To simulate the source router, we have used path combiner, which receives the packet from input ports and routes them to the target servers based on their destination. Destination routers are simulated by an output switch block which leads the packets to their destination. For each packet, destination can be one of the 1, 2 or 3 servers. Service time of servers is adjusted to 10. After processing packets in server, they will lead to *Entity Sink* or are used to measure and report the system parameters. First server output is used for measuring, $A(t)$ of the first server. The results of this simulation are shown a three axes diagram. One spots generated packet from two clients, the other shows the received packets and the last demonstrates availability of server. As shown in Figure 4, the first server was busy in 40, 50… 100… 910 simulation times, so it was idle in 10, 20, etc. (as shown Figure 4).

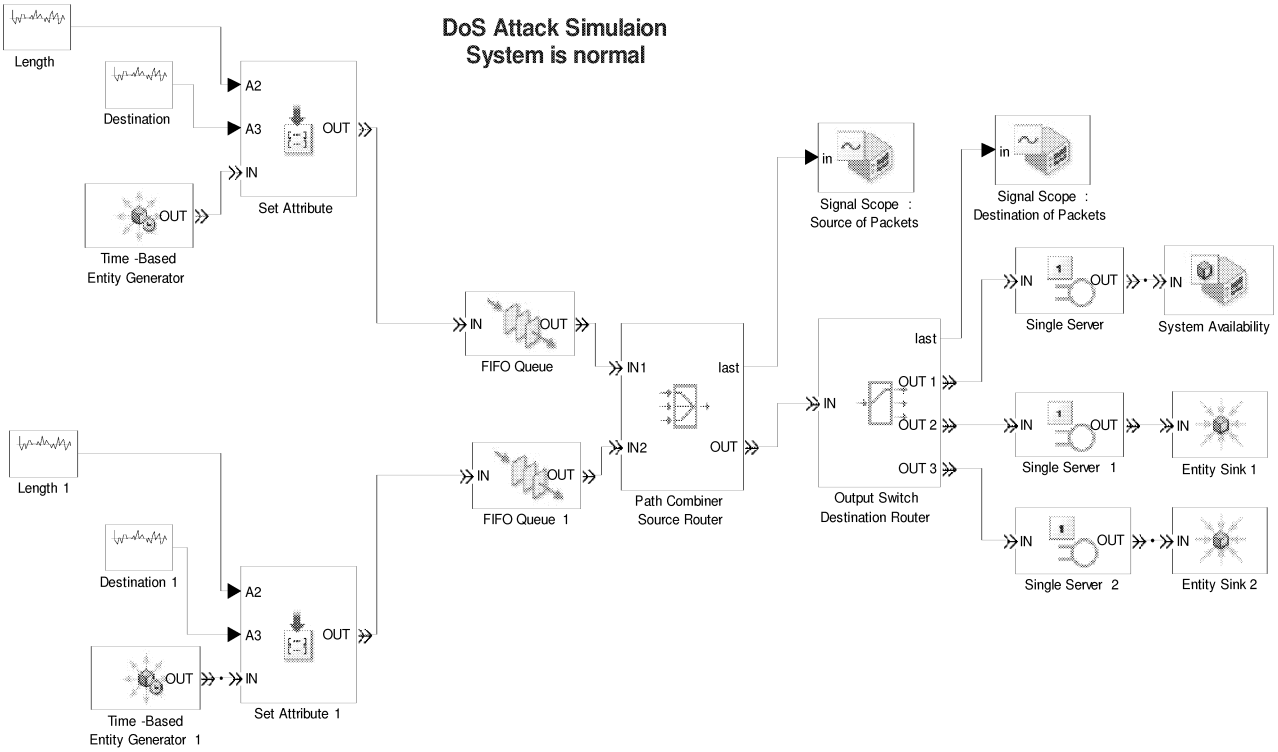


Figure 3: The System Model in the Normal State

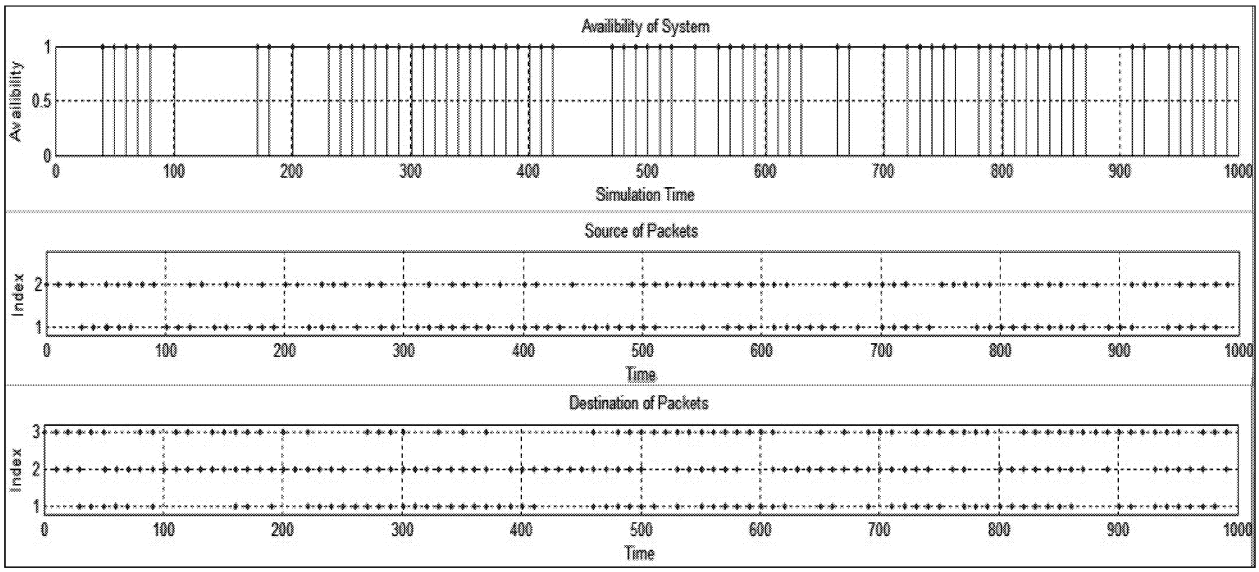


Figure 4: Simulation Results for 1000 Run-Times

Extension of the Model to Add Attacker

In this subsection, we extend the simulation model to add an attacker as a new client of the system. To model the attacker, we use a *Time-Based Entity Generator* and *Set Attribute* block, but destination of packets is not a random number and it is set to 3. Because we assumed that the aim of attacker is the third target server. In order to simulate attacks, we increase the rate of generating attacker packets such that the victim server is unable to serve and reply to the received packets.

This simulation model exposes a *DoS attack bombardment.* First, the mean value of generating packets is 6 and then by decreasing it, the victim becomes slowdown and disorder. Moreover, the output of this simulation is the number of packets departed from the attacker queue and entities departed from victim, as shown in Figure 5.

After 1000 run times of the simulation model and setting the mean value to 6, the output will be as shown in Figure 6. As the diagram says, during the total simulation time the attacker took the control of victim server such that it cannot serve to the other valid requests.

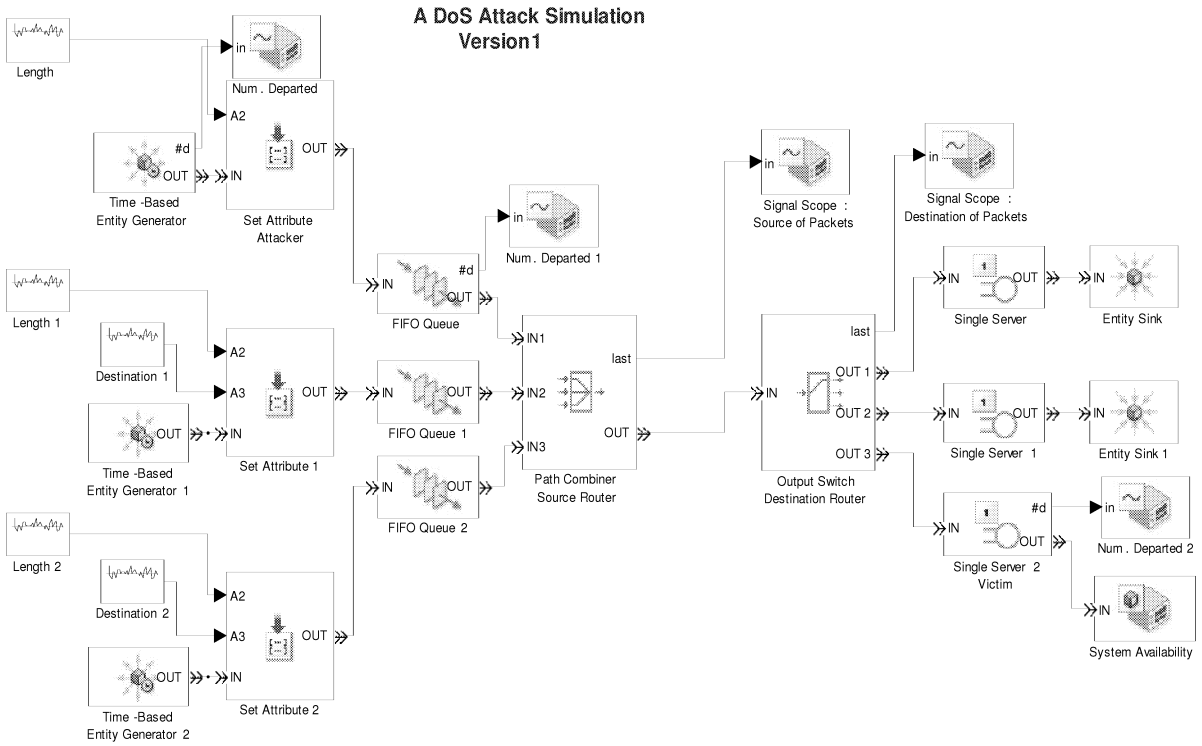


Figure 5: DoS Attack Simulation Model

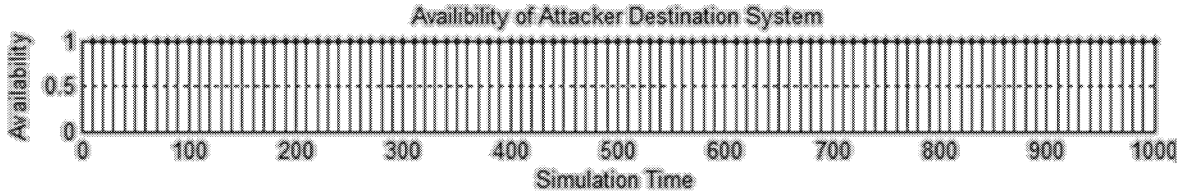


Figure 6: DoS Attack Simulation Results for 1000 Run-Times

CONCLUSIONS

In this paper, discrete-event simulation and SimEvents is used for quantitative security evaluation. The proposed model, measures the availability of system as one of security measures, before and after attack. As a case study, we modeled DoS attack as a type of packet bombardment. First, the clients send packets to servers and the attacker as a client intrudes to the system and floods packets to the victim server. During the simulation, $A(t)$ of victim server was measured.
In our future works, we intend to model and evaluate the availability measure in large and sophisticated computer and communication systems to see the potential benefits of the proposed simulation methodology for quantitative security evaluation.

REFERENCES

Stephenson, P. R. 2004. "A Formal Model for Information Risk Analysis Using Colored Petri Nets." *Proc. of the 5th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, University of Aarhus, Denmark, October 8-11.

ITSEC Webpage. "Information Technology Security Evaluation Criteria." URL: http://www.bsi.de/zertifiz/itkrit/itsec-en.pdf.

CC Webpage. "Common Criteria." URL: http://www.commoncriteriaportal.org.

SimEvents Webpage. "SimEvents", URL: http://www.mathworks.com.

Khazan, G. and M. Abdollahi Azgomi. 2008. "DNS Spoofing Attack Simulation for Model-Based Security Evaluation." *International Journal of Advanced Science and Technology*, vol. 1, no. 1, 1-8.

Khazan, G. and M. Abdollahi Azgomi. 2009. "A Distributed Attack Simulation for Quantitative Security Evaluation Using SimEvents." *Proc. of the 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'09)*, Rabat, Morocco, 10-13, IEEE CS Press

Malekian, E. 2004. *Network Intrusion and Countermeasures*. Nas Publications (in Persian)

Nicol, D. M.; S.W. Smith; and M. Zhao. 2004. "Evaluation of Efficient Security for BGP Route Announcements using Parallel Simulation." *Simulation Modeling Practice and Theory*, vol. 12, 187-216.

Nicol, D. and S. W. Smith. 2005. "Modeling and Simulation in Security Evaluation." *IEEE Security and Privacy*, vol. 3, no. 5, 71-74.

Porcarelli, S.; F. D. Giandomenico' A. Bondavalli; and P. Lollini. 2004. "Model-Based Evaluation of a Radio Resource Management for Wireless Networks." *Proc. of the 1st Conference on Computing Frontiers*, Ischia, Italy, 51-59.

Sahinoglu, M. 2007. *Trustworthy Computing: Analytical and Quantitative Engineering Evaluation*. Wiley-Interscience.

BIOGRAPHY

GOLRIZ KHAZAN received the B.Sc. degree in computer engineering (software) from Azad University of Tehran (1998) and M.Sc. degree in information technology from Iran University of Science and Technology (March 2009). Title of her M.Sc. thesis was "A Simulation-Based Framework for Quantitative Evaluation of Availability by Considering Intrusions and Malicious Attacks," which has been done under supervision of Dr. Mohammad Abdollahi Azgomi.
Mrs. Khazan's research interests include computer networks and network security. She has published several papers in international conferences.

MOHAMMAD ABDOLLAHI AZGOMI received the B.Sc., M.Sc. and Ph.D. degrees in computer engineering (software) (1991, 1996 and 2005, respectively) from Sharif University of Technology, Tehran, Iran. His research interests include performance and dependability modelling with high-level modelling formalisms such as stochastic Petri nets, tools for modelling and evaluation, verification and validation, object-oriented modelling, web services, network and web security. He has published several papers in international journals and conferences.
Dr. Abdollahi Azgomi is currently an assistant professor at the school of computer engineering, Iran University of Science and Technology, Tehran, Iran.

Video multicast using unequal error protection with Luby Transform codes

Shakeel Ahmad, Raouf Hamzaoui, Marwan Al-Akaidi
Faculty of Technology, De Montfort University, Leicester, UK.
Email:{sahmad,rhamzaoui,mma}@dmu.ac.uk

***Abstract*— In a recent work, we proposed a method for unequal error protection with Luby Transform codes and showed that it achieves lower bit error rates than a state of the art technique when the information symbols are partitioned into two protection levels (most important and least important). In this paper, we apply our previous work to the problem of video multicast with heterogeneous receivers. We provide simulations for the scalable video coding (SVC) extension of the H.264/AVC standard and show that our unequal error protection method provides significantly better objective video quality results than two state of the art techniques in applications where a high video quality is desired.**

Keywords: Channel coding, fountain codes, unequal error protection, video multicast, performance analysis.

I. Introduction

Fountain codes [1], [2], [3] are a new class of erasure codes that have two advantages over traditional erasure codes such as Reed-Solomon codes. First, they allow faster encoding and decoding, making them more suitable for real-time applications. Second, they are more flexible in the sense that the channel code rate does not have to be fixed in advance as an infinite number of encoded symbols can be built on the fly.

Luby Transform (LT) codes [2] were the first class of practical Fountain codes. Shokrollahi [3] introduced another class of practical Fountain codes called Raptor codes by concatenating a fixed-rate channel code with an LT code. Raptor codes have been adopted as enhanced application layer forward error correction (FEC) by Multimedia Broadcast/Multicast System (MBMS) of the 3rd Generation Partnership Project (3GPP), IP datacast (IPDC) of Digital Video Broadcasting - Handheld (DVB-H), as well as Digital Video Broadcasting Project's (DVB) global IPTV standard.

Recently, some works [4], [5], [6], [7] have addressed the problem of designing Fountain codes with unequal error protection (UEP) properties. In UEP, information symbols are protected according to their importance. This usually allows a better overall system performance than equal error protection where all information symbols receive the same level of protection [8].

In [7], we provided a simple method to decrease the bit error rate (BER) of LT codes by virtually increasing the number of information symbols. Moreover, we exploited this idea to propose a new technique for UEP with LT codes and compared the BER performance of our scheme to that of [4]. However, our results were provided for general data without a specific application in mind. Moreover, our UEP scheme was not compared to another state of the art technique [5]. In this paper, we apply the work in [7] to the problem of video multicast with heterogeneous receivers. We provide experimental results for the scalable video coding (SVC) [9] extension of the H.264/MPEG-4 AVC video compression standard and show that our method provides significantly better peak signal to noise ratio (PSNR) results than the state of the art techniques of [4] and [5] in applications where a high video quality is desired.

The paper is organized as follows. Section II contains background material about LT codes. Section III describes the UEP techniques of [4], [5], and [7]. Section IV presents our simulation results for the video multicast scenario.

II. Background

In this section, we explain the encoding and decoding process with LT codes.

A. Encoding

The LT encoder takes a set of k information symbols (bits or bytes, for example) and generates a potentially infinite sequence of encoded symbols of the same alphabet. Each encoded symbol is computed independently of the other encoded symbols. More precisely, given k information symbols $i_1, \ldots, i_k$ and a suitable probability distribution $\Omega(x)$ on $\{1, \ldots, k\}$, a sequence of encoded symbols $e_n,\ n \geq 1, \ldots$, is generated as follows. For each $n \geq 1$

1) Select randomly a degree $d_n \in \{1, \ldots, k\}$ according to the distribution $\Omega(x)$.
2) Select uniformly at random d_n distinct information symbols and set e_n equal to their bitwise modulo 2 sum.

The relationship between the information symbols and encoded symbols can be described by a graph (see Fig. 1 for an example).

B. Decoding

When an encoded symbol is transmitted over an erasure channel, it is either received correctly or lost. The LT decoder tries to recover the original information symbols from the received encoded symbols. We assume that for each received encoded symbol, the decoder knows the indices of the information symbols it is connected to. This is possible, for example, by using a pseudo-random

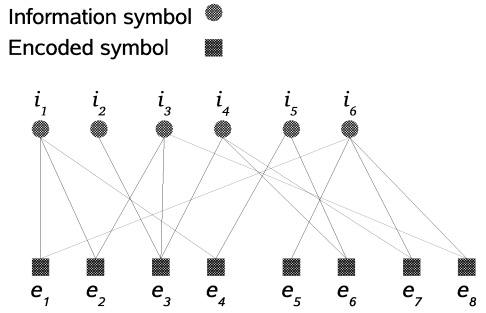


Fig. 1. Encoding graph of an LT code. Eight encoded symbols are generated from $k = 6$ information symbols. The degree of an encoded symbol is the number of information symbols that were used to generate it. For example, the degree of e_1 is equal to two.

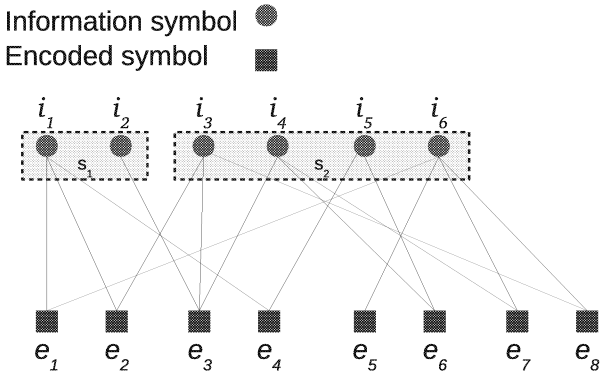


Fig. 2. UEP scheme proposed in [4]. Two levels of protection are used. The set of MIB contains two information symbols while the set of LIB contains four information symbols.

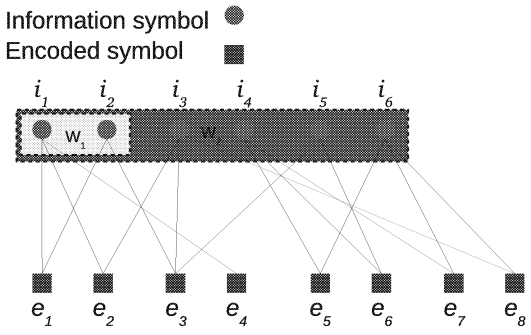


Fig. 3. UEP scheme proposed in [5]. Two levels of protection are used. The encoded symbols e_1 and e_4 are generated from the MIB class while the remaining encoded symbols are generated from the LIB class.

generator with the same seed as the one used by the encoder.

The decoding process is as follows:

1) Find an encoded symbol e_m that is connected to only one information symbol i_j. If this is not possible, stop the decoding.
   a) Set $i_j = e_m$.
   b) Set $e_x = e_x \oplus i_j$ for all indices $x \neq m$ such that e_x is connected to i_j. Here $\oplus$ denotes the bitwise modulo 2 sum.
   c) Remove all edges connected to i_j.
2) Go to Step 1.

III. Previous UEP methods

In this section, we describe the three existing UEP techniques with LT codes.

A. Rahnavard, Vellambi, and Fekri's method [4]

Rahnavard, Vellambi, and Fekri [4] were the first to propose a method to provide UEP with LT codes. For simplicity, we describe their method when two levels of protection are used. Consider a source block having k information symbols. Partition these k information symbols into two sets S_1 and S_2 of size $|S_1| = \alpha k$ and $|S_2| = (1-\alpha)k$, respectively, where $0 < \alpha < 1$. The set S_1 is called the set of most important bits (MIB) while the set S_2 is called the set of least important bits (LIB). Define probabilities p_1 and p_2 ($p_1 + p_2 = 1$) to select S_1 and S_2, respectively. Given a suitable probability distribution $\Omega(x)$ on $\{1, \ldots, k\}$, a sequence of encoded symbols e_n, $n \geq 1$ is generated as follows. For each n

1) Select randomly a degree $d_n \in \{1, \ldots, k\}$ according to the distribution $\Omega(x)$.
2) Select d_n distinct information symbols successively. To select a symbol, first select one of the two sets S_1 or S_2 (S_1 with probability p_1 and S_2 with probability p_2). Then choose randomly a symbol from the selected set.
3) Set e_n equal to the bitwise modulo 2 sum of the d_n selected information symbols. Figure 2 describes the process for two classes.

To ensure that the MIB symbols have lower BER than the LIB symbols, the probability of selecting an MIB symbol should be larger than the probability of selecting an LIB symbol [4], that is, $p_1 \frac{1}{|S_1|} > p_2 \frac{1}{|S_2|}$. To achieve this, one can set $p_1 = \frac{k_M |S_1|}{k}$ and $p_2 = \frac{k_L |S_2|}{k}$ for $0 < k_L < 1$ and $k_M = (1 - (1-\alpha)k_L)/\alpha$. Here the parameter k_M gives the relative importance of the MIB symbols.

B. Method of Sejdinovic et al. [5]

A source block having k information symbols is partitioned into L sets $S_1, S_2, \cdots, S_L$ such that the first $|S_1|$ information symbols of the source block belong to the set of most important bits, the next $|S_2|$ information symbols belong to the set of next most important bits and so on. Then L windows $W_1, W_2, \cdots, W_L$ are defined such that W_i consists of the sets $S_1, \cdots, S_i$. Thus the size of the ith window is $|W_i| = \sum_{j=1}^{i} |S_j|$ and $|W_1| < |W_2| < \cdots < |W_L|$. To generate an encoded symbol, a window is selected according to distribution $\Gamma(x) = \sum_{i=1}^{L} \Gamma_i x^i$. Here Γ_i is the probability that window W_i is chosen. Then an LT code with a suitable degree distribution is applied. UEP is achieved by choosing appropriate values for $\Gamma_1, \cdots, \Gamma_L$. Fig. 3 describes the encoding process for two classes.

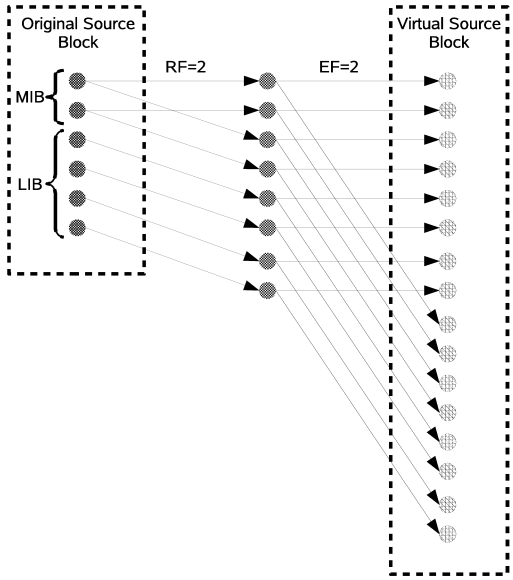


Fig. 4. Building a virtual source block with the UEP scheme proposed in [7]. Here $4k = 6$, $EF = 2$, and $RF = 2$.

C. Ahmad, Hamzaoui, and Al-Akaidi's method [7]

For simplicity and without loss of generality, we describe our UEP method on an example where there are two levels of protection (MIB and LIB symbols), $k = 6$ information symbols, and two MIB symbols. If we duplicate the two MIB symbols as in the first step of Fig. 4, the virtual size of the source block becomes 8, corresponding to a repeat factor $RF = 2$. Here RF is the number of times a block of MIB symbols is present in the virtual source block. We next extend the degree distribution of the LT code from $k = 6$ to $k = 8$. To generate an encoded symbol, we find its degree d using the new degree distribution and then select d information symbols from the 8 virtual symbols. If the index of a selected information symbol is larger than 2, we map its virtual index to the actual index by subtracting 2. This UEP technique can be combined with block duplication [7] by duplicating the virtual source block with an expanding factor EF. For example, for $RF = 2$ and $EF = 2$, the original source block consisting of two MIB symbols and four LIB symbols is transformed into a virtual block of size $EF(RF \times 2 + 4) = 16$ (Fig. 4).

IV. Experimental results

In this section, we compare our UEP technique to the UEP techniques of [4] and [5] for a video multicast scenario with heterogeneous receiver classes. In this scenario, a video server multicasts an SVC [9] encoded bitstream of $(1 + t)k$ symbols to n receiver classes. The parameter t is called the transmission overhead.

As in [6], we used the Stefan video sequence which has a spacial resolution of 352×288 and a temporal resolution of 30 fps. The first group of pictures (GOP) of the sequence was encoded using the SVC reference software (JVSM) into one base layer (BL) and 14 enhancement layers (ELs). The GOP had a length of 16 frames. The resulting layer sizes, bit rates, and PSNR are summarized in Table I. A source block consisted of one GOP, and its size was about 190,000 bytes.

Decoded layers	Size	Bitrate	PSNR
BL	400	292.37	25.79
BL + 1 EL	700	510.65	27.25
BL + 2 EL	875	636.56	28.14
BL + 3 EL	1155	839.82	29.00
BL + 4 EL	1550	1127.10	29.51
BL + All ELs	3800	2764.55	40.28

TABLE I
SVC encoding of the first GOP of the Stefan video sequence (352x288,30 fps) into one base layer (BL) and 14 enhancement layers (EL). The table shows the number of symbols, the bitrate in kbps, and the Y-PSNR in dB.

As in [6], each symbol was equal to 50 bytes, giving $k = 3800$ symbols per source block. In each source block, the base layer (containing 400 symbols) was chosen as the MIB set, whereas the remaining symbols built the LIB set. The video server transmitted 250 source blocks. At the receiver side, we assumed that a layer can be used to enhance the video quality only if it was decoded fully, and all the layers before this layer were also decoded fully. In this way, the number of consecutively decoded information symbols, starting from the first information symbol, determined the number of decoded layers.

For the UEP scheme of [4], we followed the recommendation in [4] and used $k_M = 2$ and the fixed degree distribution of [3] for both the MIB and LIB sets. This degree distribution is given by

$$\begin{aligned}\Omega(x) =& 0.007969x + 0.493570x^2 + 0.166220x^3 + \\ & 0.072646x^4 + 0.082558x^5 + 0.056058x^8 + \\ & 0.037229x^9 + 0.055590x^{19} + 0.025023x^{64} + \\ & 0.003135x^{66}.\end{aligned} \tag{1}$$

For the UEP scheme of [5], we followed the recommendation in [6] and used $\Gamma_1 = 0.11$, the fixed degree distribution (1) for the LIB class, and the robust soliton distribution [2] with parameters $c = 0.03$ and $\delta = 0.5$ for the MIB class.

For our UEP scheme, we used the robust soliton distribution with parameters $c = 0.01$ and $\delta = 0.5$.

Fig. 5 shows the PSNR performance of our UEP scheme as a function of the symbol loss rate for various settings of RF and EF. Here the transmission overhead t was equal to one, and the video was transmitted to one receiver. The results show that increasing RF from 2 to 3 increases the likelihood of decoding the BL successfully, but decreases the likelihood of successfully decoding the ELs, leading to a decrease of the overall PSNR. Increasing EF increases the average degree and leads to duplicate selection of the information symbols, which decreases the

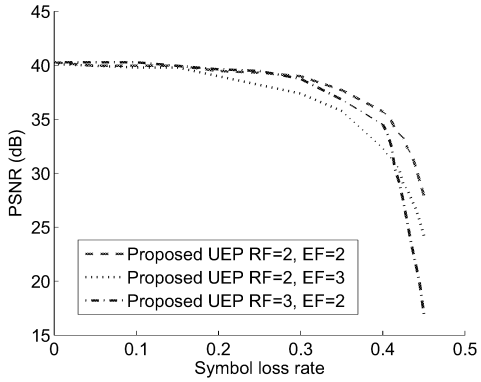


Fig. 5. PSNR as a function of the symbol loss rate for the transmission of the Stefan sequence with the proposed UEP scheme. The performance of the scheme is shown for different settings of EF and RF.

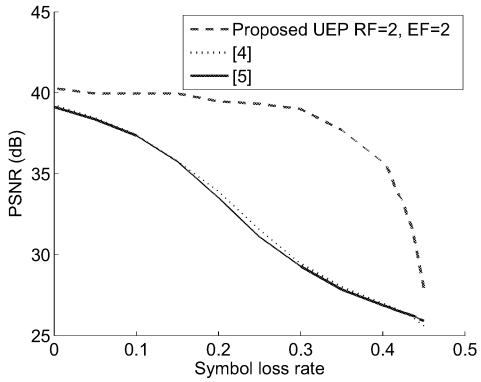


Fig. 6. PSNR as a function of the symbol loss rate for the proposed UEP scheme, the UEP scheme of [4], and the UEP scheme of [5].

performance. Based on these results, we set $RF = 2$ and $EF = 2$ for the following experiments.

Fig. 6 shows the PSNR perceived by a receiver class as a function of the symbol loss rate for our UEP scheme, the UEP scheme of [4], and the UEP scheme of [5]. Our scheme was more robust to increasing symbol loss rate.

Fig. 7 shows the average PSNR performance for a multicast transmission to $n = 5$ receiver classes with symbol loss rates 0, 0.05, 0.1, 0.15, and 0.2. Our UEP scheme always outperformed the UEP scheme of [4]. It also outperformed the UEP scheme of [5] over the range of transmission overheads that are required to provide full decoding of the BL. In particular, our UEP scheme achieved an average video quality of 36 dB with 60 % less transmission bandwidth.

Fig. 8, 9, and 10 show the individual PSNR results for receiver classes with symbol loss rates 0, 0.1, and 0.2, respectively. Compared to our UEP scheme, the scheme of [5] achieved an acceptable video quality (25 dB) for the receiver with the worst channel conditions at a lower overhead (Fig. 10). However, with the scheme of [5], the maximum PSNR for this receiver did not exceed 32.5 dB even at $t = 1$. In contrast, our scheme reached this PSNR for a transmission overhead t=0.48, exceeding 35 dB at

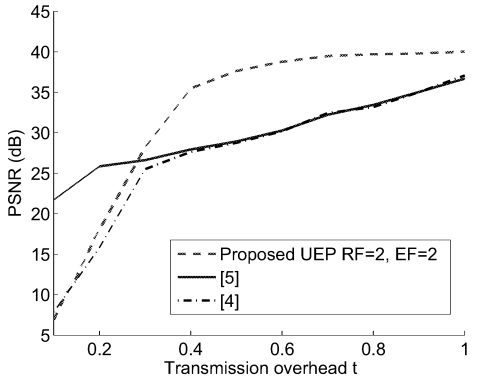


Fig. 7. Average PSNR of $n = 5$ receiver classes as a function of the transmission overhead.

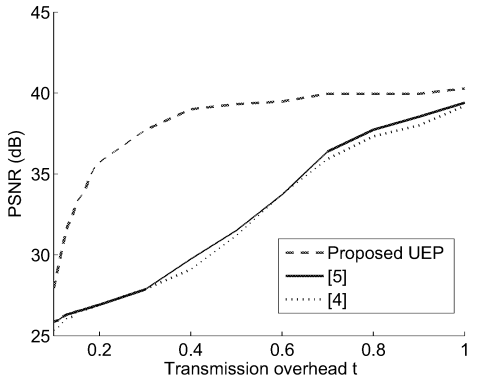


Fig. 8. PSNR of receiver class with symbol loss rate 0 as a function of the transmission overhead.

$t = 1$.

V. Conclusion

We used our LT-based UEP scheme [7] for video transmission over a lossy channel. Simulations for the SVC extension of the H.264 standard showed that our scheme can provide up to 7 dB improvement in PSNR over the UEP schemes of [4], [5] for unicast transmission. For multicast transmission to a set of receivers with different channel conditions, simulations showed that our UEP scheme has a better average PSNR performance when the transmission overhead is large and a worse performance when the overhead is low. This makes our UEP scheme more appropriate in applications where high video quality is desired. In this paper, the values of RF and EF were optimized by simulations. Future work could be the development of faster optimization techniques based on analytical models. Other future work could extend the proposed framework to the situation where the information symbols are partitioned into more than two sets.

VI. Acknowledgment

This work was supported in part by the DFG Research Training Group GK-1042. We thank Vladimir Stanković for providing a draft of paper [6].

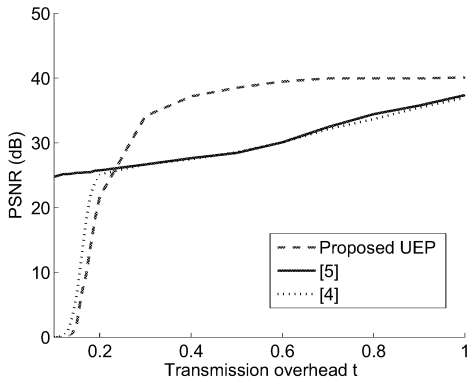


Fig. 9. PSNR of receiver class with symbol loss rate 0.1 as a function of the transmission overhead.

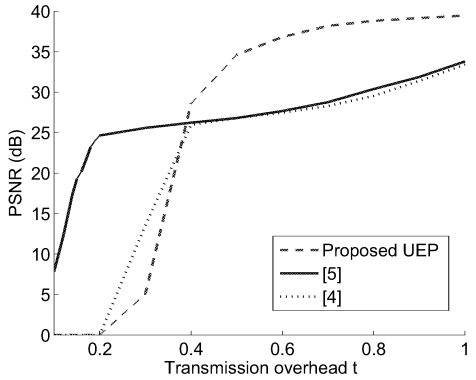


Fig. 10. PSNR of receiver class with symbol loss rate 0.2 as a function of the transmission overhead.

REFERENCES

[1] J.W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast", IEEE Journal on Selected Areas in Communications, vol. 20, pp. 1528–1540, Oct. 2002.

[2] M. Luby, "LT-Codes", Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 271–280, 2002.

[3] A. Shokrollahi, "Raptor Codes", IEEE Trans. Inf. Theory, Vol. 52, No. 6, June 2006, pp. 2551–2567.

[4] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property", IEEE Trans. Inf. Theory, Vol. 53, No. 53, pp. 1521–1532, April 2007.

[5] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding window fountain codes for unequal error protection", Proc. 41st Asilomar Conf., Pacific Grove, pp. 1020–1024, 2007.

[6] D. Vukobratovic, V. Stanković, D. Sejdinovic, L. Stanković, and Z. Xiong, "Scalable video multicast using expanding window fountain codes", IEEE Transactions on Multimedia, Special Issue on Quality-driven Cross-layer Design for Multimedia Communications, vol. 11, August 2009.

[7] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Unequal error protection using LT codes and block duplication", Proc. MESM 08, Middle Eastern Multiconference on Simulation and Modelling, Amman, Aug. 2008.

[8] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bitstreams, IEEE Signal Proc. Mag., vol. 22, pp. 91-107, Nov. 2005.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Trans. Circuits and Systems for Video Tech. vol. 17, pp. 11031120, September 2007.

BIOGRAPHIES

Shakeel Ahmad received the B.Sc. degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2000, the M.Sc. degree in Information and Communication Systems from Hamburg University of Technology (TUHH), Germany, in 2005, and the Dr.-Ing degree from the University of Konstanz, Germany, in 2008. He is currently working as a Research Fellow with the Faculty of Technology at De Montfort University, Leicester, U.K. His research interests include multimedia communication, error-resilient video streaming and channel coding.

Raouf Hamzaoui received the Maîtrise de mathématiques from the Faculty of Sciences of Tunis, Tunis, Tunisia, in 1986, the M.Sc. degree in mathematics from the University of Montreal, Montreal, QC, Canada, in 1993, the Dr. Rer. Nat. degree from the Faculty of Applied Sciences, University of Freiburg, Freiburg, Germany, in 1997, and the Habilitation degree in computer science from the University of Konstanz, Konstanz, Germany, in 2004. He is currently a Professor of Media Technology in the Faculty of Technology at De Montfort University, Leicester, U.K. His research interests include image and video compression, multimedia communications, error control systems, and algorithms.

Marwan M. Al-Akaidi (PhD CEng CIP FIEE FBCS FIAP SMIEEE) is a Professor and research group leader of the Mobile & Wireless Multimedia Comms & SP group. He obtained his first degree (82) in Electronic Engineering (UK) and his MSc & PhD in Digital Communications in 84 & 88 respectively from Loughborough University, UK. He led the school of Engineering & Technology at De Montfort University from Jan. 2002 to Sept. 2008, and currently is appointed as the Director of External relations. He enjoys strong links with UK & Europe industries including Siemens, O2, BT, Orange, Nokia, BMW, Chronos and many others. He is interested in all aspects of Mobile & Wireless Communications & signal processing (including Speech & video) and he is the author of the "Fractal Speech Processing" book (Cambridge press). He is the SPC chair & conference coordinator of the IEEE UKRI. He is also the chair of the IET 3G Technology & beyond 3G technology conferences as well as the WiMAX08 London, WiMAX09 London conferences and the chair & founder of the MESM conferences since 99. He is also editor-in-chief of AWMC & editorial member of many national & International Journals.

COMPLEX SYSTEMS SIMULATION

TOWARDS AN OPTIMIZATION PROBLEM FOR THE DESIGN OF PORTS

Ghassan ELCHAHAL
Marine Department
COWI A/S
Doha-QATAR
ghel@cowi.com

Rafic YOUNES
Lebanese University,
Rafic Harriri Campus,
Beirut, LEBANON
ryounes@ul.edu.lb

Pascal LAFON
Institut Charles Delaunay
University of Technology of Troyes
Troyes , FRANCE
pascal.lafon@utt.fr

KEYWORDS
Numerical Modeling, Marine Engineering, Shape optimization.

ABSTRACT
A shape optimization problem has been developed for a detached breakwater used in port expansion projects. Most often, a detached breakwater seems to be an efficient solution for its several advantages. But, these marine structures are characterized by high costs due to the type of used equipments during construction. Therefore, an optimum shape of these breakwaters, to economize its cost, constitutes an advanced step in their design. The detached breakwater is parameterized by the coordinates of its extremity points and it is free to move in the computational domain. The optimization problem seeks to determine the optimum shape and location of this structure in consistent to the design constraints. These latter are expressed by a maximum allowable wave disturbance inside the port and a satisfactory clearance distance at the port entrance.

INTRODUCTION
In this decade, we are witnessing expansion projects for the existing ports in many countries. Some of these projects require the construction of another coastal breakwater where others require the addition of a detached or offshore breakwater. This may differ from one site to another due to the geographical location of the port, the wave climate in the relevant region, and the functional type of the port itself or mainly the vessel size. The detached breakwater constitutes an essential opportunity to be considered by the designers in such expansion projects. It has been also used in LNG terminals that are expanding quickly due to the worldwide increasing demand of the Gas. For example, moving the breakwater seawards allows more space for accommodating splash due to wave overtopping behind the breakwater and this significantly lowers the effects of overtopping at the limit of the working area within the harbor. This can also means that the crest level of the structure can be reduced (CIRIA C683, p 787). A detached breakwater is likely to be shorter than a shore-connected one; also the volume of the material needed will be less because of the reduced length. But on the other side the cost of the construction maybe higher because waterborne equipment has to be used. This ensures the importance of an optimization problem for such types of engineering problems.

It is noticed that applications related to hydrodynamic aspects of marine structures are rarely reported. This may result from several severe problems related to the hydrodynamic analysis and evaluation of such structures and mainly summarized in the difficulty of hydrodynamic analysis for their arbitrarily shapes. Nevertheless, there is some work spent on optimization in the ocean field but only for offshore structures; for example Chou (1977) derived optimal shapes for a buoy and an ocean platform supported by four columns proposing an analytical procedure, Akagi & Ito (1984) optimized the heave motion of a hydrodynamic transparent semi submersible using a quadratic programming technique, Kagemoto (1992) optimized the arrangement of vertical floating cylinders in waves, Clauss & Birk (1996) focused on hydrodynamic shape optimization for large offshore structures (oil platforms) based on non linear programming algorithms. Elchahal et al. (2008) addressed an optimization problem for floating breakwaters.

To our knowledge, shape optimization techniques are almost never used in coastal engineering problems except by Isebe et al. (2008). Groynes, breakwaters, jetties and many other marine structures have been designed using simple hydrodynamical assumptions, structural strength laws, and empirical considerations. Isebe et al. (2008) worked on the optimization of defense structures to minimize free surface elevation for fixed bathymetry coastal areas. They tried to find the optimum dimensions of groynes that yield to a minimum wave height.

The main objective of this paper is to develop an optimization problem for the design of a detached breakwater used in the ports expansion projects. In other words, find the optimum shape of the detached breakwater and in the optimal location, respecting the wave height and other geometrical constraints inside the port. The idea reverts to the fact that in such fluid-structure interaction problems, since the flow is affected by the presence of a structure in the studied domain; then for sure the structure shape and location can be modified to attain a targeted flow.

PROBLEM DEFINITION

In Engineering design problems, it is the designers' task to specify the variables of the problem utilizing their knowledge and experience The designer may develop different port layouts that meet the same functional requirements. On the other side, significant cost savings can usually be made by minimizing the breakwater length. It can be noted that computational or physical modeling studies are on average equivalent to the construction of 1 t o 2 meters of breakwater and savings can be equivalent to the cost of several tens of meters of breakwater (CIRIA manual). Therefore, an optimization problem will be an essential opportunity to determine the length of the detached breakwater and its best location. In some cases, it can replace the physical modeling totally, or it can constitute a preliminary design for the physical modeling tests.

The optimization problem is introduced and explained in section 3. It is composed from an objective or cost function, defined by the minimum length of the detached breakwater, and the relative constraints: Restricted wave height at a desired location inside the port and minimum opening distance at the entrance of the port. When looking for the best location and shape for a detached breakwater it is advisable to study the bathymetry and also the possible refraction, diffraction, and reflection inside the port. Wave assessment inside the port requires a numerical modeling of the wave's propagation from the ocean towards and inside the port area. This hydrodynamic study of the waves is explained in section 4. Finally, the modeling results are presented in section 5.

OPTIMIZATION PROBLEM

A shape optimization problem consists in the minimization of a functional $f \in R$, also called cost or objective function, corresponding to the physical criteria to optimize. This function depends on x_i, a design parameter defining the shape within the admissible set *X*, also called control space. In the application presented here, a given set of parameters x_i defines a new shape for the structure, so a new domain $\Omega(x_i)$ for the computation. Then by using the model presented in next sections, we compute the length of the detached breakwater and thus, the value of a cost function f. Also, we compute the free surface elevation η in Ω to determine the wave height at a desired location inside the port depending of the solution η.

Parameterization of the structure

We aim to prove that the surface elevation of water waves inside the port can be monitored by the shape of the detached breakwater and its length. We parameterize the structure by using the coordinates of the two points at the extremities of the detached breakwater, (Figure 1). These two points are free to move in the domain of computation to attain the optimal result for the port layout. Therefore, the numbers of variables will be 4, $x_i = [x_1, y_1, x_2, y_2]$

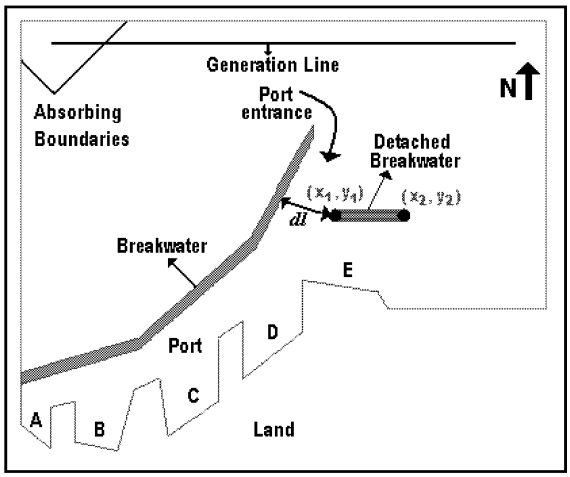


Figure 1 Parameterization of the structure and domain definition

The optimization problem is assumed to be finite dimensional constrained minimization problem, which is symbolically expressed as:

Find a design variable vector x_i;

to minimize the cost function $f(x_i)$

subjected to the *n* constraints $C_n(x_i) \le 0$

Bathymetry definition

The bathymetry of the port is presented in **Figure 2**. It is observed that the water depth varies from -30m to -5m inside the port, where it is around -40m far in the seaside.

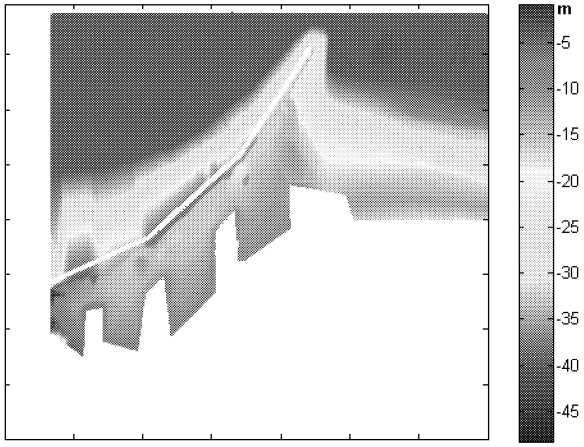


Figure 2 Bathymetry definition of the port

Objective Function

The optimal solution is to design the detached breakwater with a minimum length and an optimum location inside the port. This will minimize the construction costs and at the same time will improve the overall functionality of the expanded port. It is expressed in terms of the geometrical dimensions of the structure by:

$$f_{ob}(x_i) = Min\left[(x_2 - x_1)^2 + (y_2 - y_1)^2\right]^{1/2}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of the representative points of the detached breakwater.

Wave Height Constraint

The wave height distribution inside a harbour is an important consideration in planning the layout of the

facilities prior to construction. After expanding the port, the new quay E must have the also wave height restrictions to enable functional operations like the other Quays A, B, C, D. This constraint is expressed in terms of the surface elevation at a desired point at the Quay E.

The construction of wave dissipating devices (i.e. breakwaters) around man made works on the coast must be considered to reduce wave-height to an acceptable level. In particular, the criteria for acceptable wave action at a harbour vary from one port to another due to the type and size of vessels using this port. The allowable wave height constraint is expressed as follows:

$$C_1(x_i) = \eta(x, y, t) < a$$

Where x, y represent the coordinates of a desired point in the computational domain, and a is the maximum admissible value of the wave height at the desired position. (a is considered to be equal to 0.3m)

Port Entrance Constraint

The location of the new detached breakwater should not block the entrance of the port. The ships must still have enough space to sail in and out. The distance between the extremity of the detached breakwater at the point (x_1, y_1) and the existing one must respect a minimum distance identified for each port as required due to its functional type.

$$C_2(x_i) = \left[(x_1 - x_m)^2 + (y_1 - y_m)^2\right]^{1/2} > dl$$

Where x_m, y_m represents the coordinates of the end point of the perpendicular line from the point (x_1, y_1) to the detached breakwater, and dl is the minimum allowable clearance distance at the port entrance. (dl is considered to be equal to 60m)

HYDRODYNAMIC MODELLING

Fluid is assumed to be ideal, flow is considered as irrotational, so we can apply a linear wave theory. An incident monochromatic wave is characterized by its height H, its wave period T, its wavelength L, angular frequency ω=2π/T, and its direction θ. The incident velocity potential of linear waves propagating is given by:

$$\phi_I = -\frac{igA}{\omega}\frac{\cosh\left[k(z+d)\right]}{\cosh(kd)}e^{ikx}$$

where g is the gravitational acceleration, d is the water depth, A is the wave amplitude and k is the wave number satisfying the dispersion relation,

$$\omega^2 = gk\tanh(kd)$$

A numerical wave model is required to study the propagation of waves from deep water towards the area of interest inside the port. The well known mild slope equation, derived by Berkhoff (1976), for linear wave propagation has been adopted. It combines the resolution of the three main physical mechanisms producing wave transformation: diffraction, reflection and refraction. In general, the diffraction process is of primary importance at the tips of groynes, breakwaters or harbor entrances. As harbors are semi-closed wall surrounded areas, the reflection mechanism must also be taken into account. Furthermore, the refraction effect plays a secondary role at areas with a relatively flat bottom, but is very relevant when passing over dredged navigation channels. The mild slope equation is expressed in its elliptic form as follows:

$$\nabla(C.C_g.\nabla\phi) + k^2 C.C_g.\phi = 0$$

Where ϕ is the velocity potential, C and C_g are phase and group velocity.

$$C = \sqrt{\frac{g}{k}\tanh kd} \qquad \text{and} \quad C_g = C\frac{(1 + \frac{2kd}{\sinh 2kd})}{2}$$

Solving Berkoff's mild slope equation allows obtaining the instantaneous velocity potential inside the fluid domain. Then, the free surface elevation with respect to the mean water level at each point within the harbour is easily computed, by the following relation:

$$\eta(x, y, t) = \frac{i\omega}{g}\phi(x, y, t)$$

The Berkoff equation is associated with the following boundary conditions:

Reflection Condition

The breakwaters and the Quay walls, marked by blue colour in Figure 1, represent rigid structures. Hence, the flow is reflected on these structures. The fully reflective boundary condition is really inappropriate since in reality partial reflection at the boundaries within a port invariably occurs. Moreover, reflecting boundaries are not always vertical nor are fully reflecting. In order to treat these boundaries in the present model, they are schematized as partially reflective and introduced by using a mixed boundary condition instead of the full reflection condition in the manner proposed by Berkhoff (1976).

$\frac{\partial\phi}{\partial n} = ik\frac{k_r - 1}{k_r + 1}\phi$, where k_r represents the reflection coefficient ($k_r = 0.3$)

This boundary coefficient is applied at all the blue coloured lines in Figure 1 (24 boundaries).

Radiation Condition

The red lines in Figure 1, represent a transparent open sea boundary where it should not generate any parasitic scattered waves in the computational domain. All the flow approaching these boundaries will be absorbed outside the domain. It is expressed in the following form (Sommerfield outward radiation condition):

$$\frac{\partial\phi}{\partial x} \pm ik\phi = 0 \text{ for } x \mp \infty$$

This radiation condition is applied to the red coloured lines in Figure 1. (4 boundaries)

This boundary value problem is solved using the finite element method. The boundary conditions are expressed in terms of Dirichlet ($h.\phi = r$) or Newmann ($\vec{n}.(c\nabla\phi) + q.\phi = g$).

APPLICATION & RESULTS

The port is almost sheltered from the western waves, but it is exposed for waves coming from the north, north-east, and north-west. In Table 1, the wave conditions are defined by their Height (H) and corresponding period (T).

The computations were performed for two cases as shown in the table below.

Wave Direction (°)	H (m)	T (s)
N (0°)	2	9
NW(315°)	4	10
NE (45°)	3	10

Table 1 Wave conditions

The layout has a direct effect on the design of port and also on the design of the marine structures themselves. Thus, alternative layout options should be analyzed which will multiply the different options or possibilities of the design.

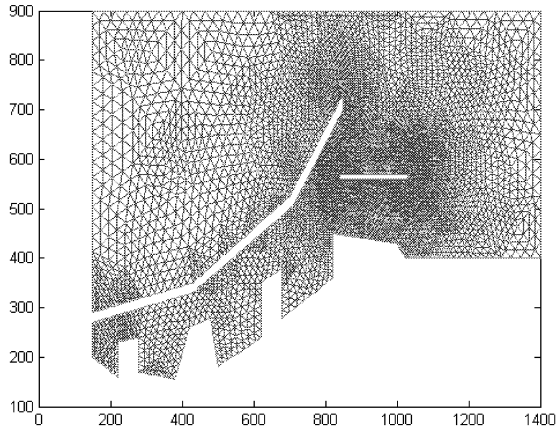


Figure 3 Computational domain including the detached breakwater in its initial form

The aim of the optimization problem is to determine the set of variables, $x_i = [x_1, y_1, x_2, y_2]$, in order to determine the optimal solution of port protection. An initial feasible solution is required to initialize the optimization procedure. It is described by a horizontal breakwater above the Quay E that must be protected (Figure 3).

Numerical implementation

A numerical solution procedure of Berkhoff's model was implemented in MATLAB™ using the Finite Element Method. The optimisation process is carried out with the SQP routine of MATLAB™.

The optimization approach is characterized by a highly non linear problem. This mainly reverts to the hydrodynamic constraint represented by solving the Berkoff equation. Each evaluation of the objective function and constraints of the problem requires a FEM simulation. A new port layout is defined in each iteration due to the new coordinates of the breakwater extremities. The time cost is almost the same for each iteration of the optimization problem, since the global model is fixed but the place and size of the detached breakwater varies in each iteration.

The optimization problem has resulted in the following coordinates which represent the location and length of the detached breakwater:

For the first case where the northern waves are applied, the optimization algorithm yields to the corresponding layout Figure 6. In order to show the significance of the optimal breakwater solution, the same computation has been carried out for the initial solution as shown in Figure 5 (horizontal detached breakwater).

x_1	y_1	x_2	y_2
830	610	1100	530

The optimal solution has resulted in an inclined breakwater and with longer length than the initial solution. This has minimized the effects of diffraction that were clearly observed around the horizontal breakwater in Figure 5. The wave height is reduced some values around 0.1m instead of 0.2m for the horizontal breakwater. Moreover, the Quay E is well sheltered in comparison with Figure 4, where the wave height attains 0.8m around the Quay E.

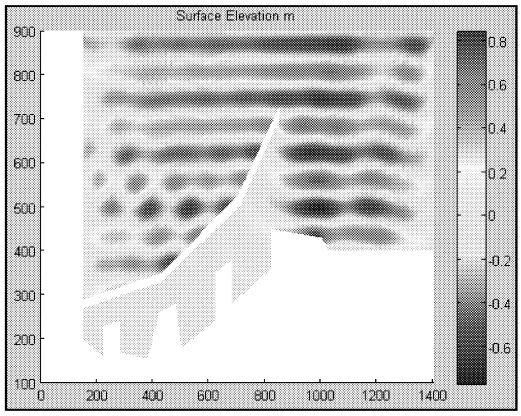


Figure 4 Modeling for Northern waves excluding the detached breakwater

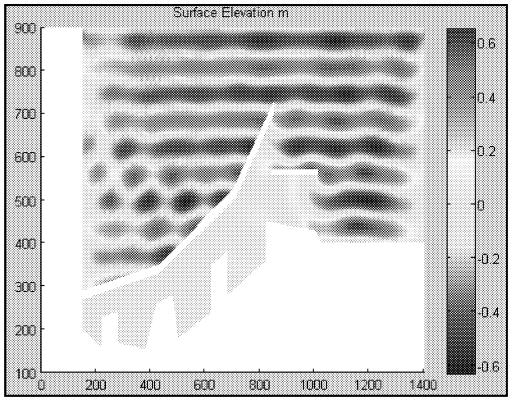


Figure 5 Modeling of the initial solution for Northern waves

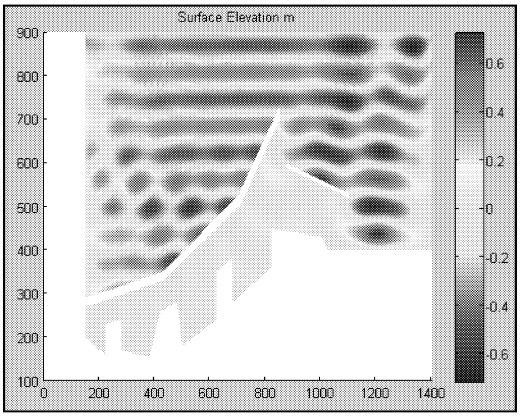


Figure 6 Optimization results for Northern waves

On the other side, the inclined detached breakwater has protected the eastern side of the Quay E from the high waves created by the diffraction of the northern waves in the case of the straight breakwater. The North-West waves are influencing the operations inside the harbour. This can be easily noticed in Figure 7, where the waves impinge Quay E with a height of 0.35m. With the help of the detached breakwater, it has been reduced to around 0.15m (Figure 8).

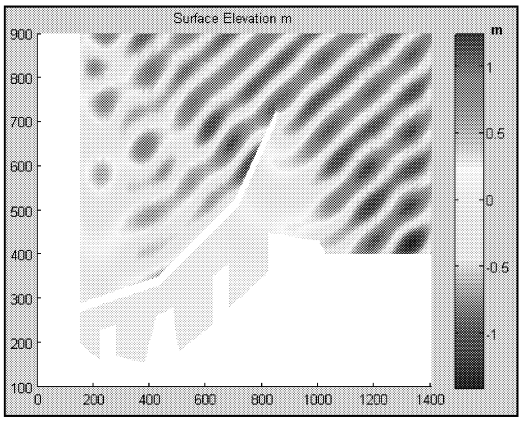


Figure 7 Modeling of North -West waves without the detached breakwater

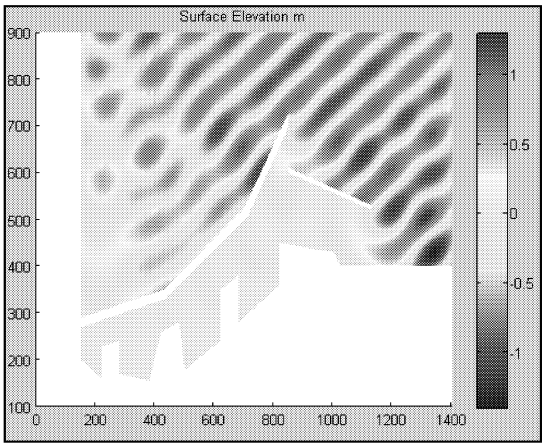


Figure 8 Modeling of North-West waves including the optimal solution

For the North-East waves, the optimal breakwater seems to constitute an acceptable solution to attenuate these waves. They have been reduced to about 0.2m.

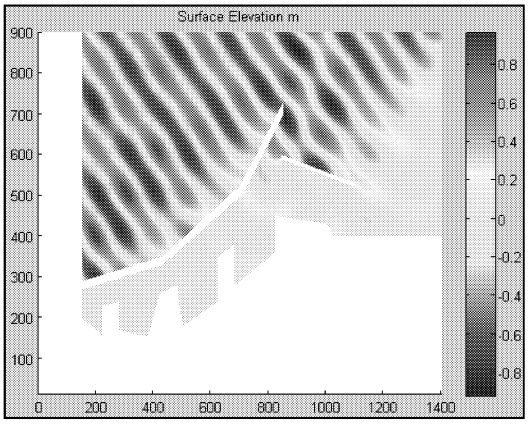


Figure 9 Modeling for the North -East waves with the optimal solution

CONCLUSION & PRESPECTRIVES

An optimization problem has been derived for the design of ports. The aim was to find an optimal layout for the port with an additional detached breakwater. This latter was described by a straight line inside the computational domain and represented by the coordinates of its extremities. The optimal solution yielded an inclined breakwater, where noticeable reduction is marked in comparison with the case of initial solution and the case of breakwater absence. This proves the importance of the optimization approach in the design of ports, but some improvements can be proposed, for future research, to attain more realistic solutions.

For example, the representation of the detached breakwater can be ameliorated by adding the number of points describing the structure itself. Increasing the number of variables to 6 (3 points) and 8 (4 points) will have an essential effect on the optimal shape of the breakwater. A larger number of points yield to more flexibility in attenuation by forming new geometrical shapes instead of the straight line breakwater.

Also, the bathymetry can be introduced to approach a more practical application. Then, the waves will be subjected to shoaling and refraction over the varying bathymetry.

Finally, almost all the coastal countries have already developed their ports and the attention is now focused on the expansion projects. The elaboration of this optimization problem will be an essential element to achieve the progress in designing the expansion of these ports in the future.

REFERENCES

Akagi, S. & Ito, K., “Optimal design of semisubmersible form by minimizing its motion in random seas”. J. Mechanism, Transmissions& Automation in Design, Trans ASME, 106 (1984) 23-30.

Chou, F.S., “A minimization scheme for the motions and forces of an ocean platform in random seas”. SNAME Trans., 85 (1977) 32-50.

CIRIA C683. "The Rock Manual, The use of rock in hydraulic engineering" 2nd edition, London (2007).

Clauss, G.F. & Birk. L. “Hydrodynamic shape optimization of large offshore structures”, Applied Ocean Research 18 (1996) 157-171

Elchahal G., Lafon P., Younes R. "Multidisciplinary optimization for a moored floating breakwaters" 18th International Offshore and Polar Engineering Conference (ISOPE-California-USA), July 2008 , Vancouver – Canada .

Isebe D., Azerad P., Mohammadi B., Bouchette F., "Optimal shape design of defense structures for minimizing short wave impact", Coastal Engineering 55 (2008) 35-46.

Kagemoto, H., “Minimization of wave forces on an array of floating bodies”. Appl. Ocean Res., 14 (1992) 83-92

EVALUATION OF THE SOFT ERROR RATE OF A SPACE APPLICATION EXECUTED BY A REAL TIME OPERATING SYSTEM

Paul PERONNARD, Esteban CABANILLAS, Raoul VELAZCO
TIMA Laboratory
46, av. Felix Viallet, 38031, Grenoble, France

Email: firstname.lastname@imag.fr

Wassim El FALOU, Haissam ZIADE
LASTRE Laboratory
Faculty of Engineering I, Tripoli, Lebanon

Email: wafalou@ul.edu.lb; hziade@ul.edu.lb

KEYWORDS
SEU, soft errors, fault injection, space application, RTOS, LEON3 processor

ABSTRACT

The soft error rate of a space application, an AOCS (Altitude and Orbit Control System) was evaluated by means of fault injection sessions. This application was executed with a matrix multiplication program both running in parallel on a real time operating system. Obtained results are compared with those issued while the application is executed in standalone mode. Experiments were performed on a LEON3 processor based architecture using a test platform devoted to radiation ground testing.

I. INTRODUCTION

The increasing sensitivity of integrated circuits to the effects resulting from the interaction with natural radiation present in the space environment or in the Earth's atmosphere, lead designers and final users to take care of these effects in order to insure the reliable operation of the system. Indeed, an energetic particle (heavy ion, neutron, proton,) hitting a critical area of the system may induce failures with potentially serious consequences. Among the effects of radiation on integrated circuits, the Single Event Upset phenomenon (SEU) is considered critical as its consequence is the modification of the content of a memory cell randomly in time and location (Normand 1966) (Ma and Dressendorfer 1989).

At the system level an SEU, also called bitflip, may propagate with different kinds of consequences depending on the criticity of the final application. This makes the evaluation of the error rate of the system a mandatory task. To achieve such an evaluation, usually adopted strategies are based on the results of accelerated tests, also called radiation ground testing. These tests are performed in facilities such as cyclotrons or linear accelerators and consist in exposing the considered circuits to suitable particle beams, while they perform an activity considered as representative of the one they will carry out in the final application (Koga et al. 1985).

Operating systems are used in many space applications. Some examples are: Herschel and Planck Mission, LISA Pathfinder or NASA ST5 satellite. In the case of the Herschel and Planck Mission, the on-board architecture is based on ERC32 processor modules, with each module having a spare (cold redundant backup) providing a total of 4 ERC32 processors. RTEMS is the used RTOS (Real Time Operating System) for the execution of the Central Data Management Unit software and the Attitude Control Computer software. In such heterogenous software systems, it is difficult to estimate the error rate by means of accelerated tests because of both the complexity and the difficulty of getting informations about the instant of occurence and the location of the SEUs resulting of particle hits in a device of the system. Moreover, any change in the software executed by the processor implies the realization of extra radiation ground testing campaigns, increasing the development costs.

Fault injection offers a promising alternative to radiation ground testing. Indeed, in (Peronnard et al. 2008b) it was proven that fault injection performed on a dedicated platform in which the errors are simulated by HW/SW means, may contribute to accurately predict the error rate.

In the present paper are presented preliminary results of fault injection sessions performed on an operating system executing in parallel two applications running on a LEON3 processor. The first one is a software issued from a real space application, a AOCS (Attitude & Orbit Control Subsystem) module and a complex processor (the Leon processor) and the second one is a benchmark program based on a matrix multiplication algorithm.

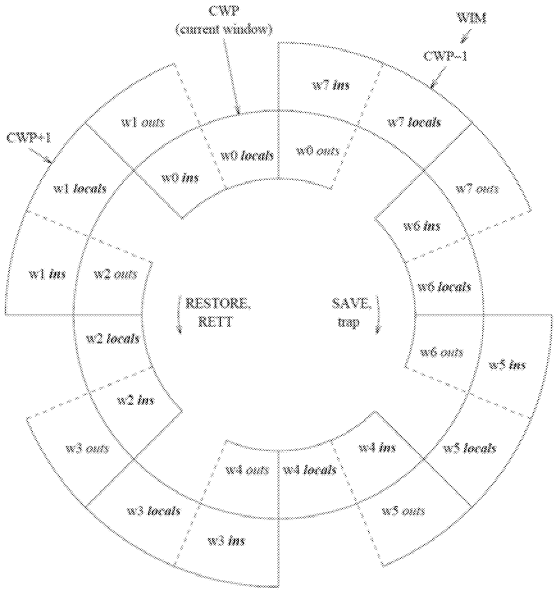
In section II are described the whole HW/SW setup and the test methodology used in this work. The software environment as well as the approach and the needed tools used to inject SEU faults in a complex application executed by the target processor are detailed. In section III are discussed results issued from fault injection experiments. In section IV are summarized conclusions and future work.

II. FAULT INJECTION SETUP

A. Device and Software Under Test

Device under test: The device under test used throughout this study is a LEON3 processor implemented on a Xilinx Virtex4 FPGA. The processor, provided by Gaisler Research (Gaisler), is a 32bit synthesisable processor core based on the SPARC V8 architecture (SPARC 1992). It runs at a

frequency clock of 66MHz. The main characteristic of this processor is the use of a large "windowed" register file. At any given instant, a program sees 8 global integer registers plus a 24 register window into a larger register file. The windowed registers can be described as a cache of procedure arguments, local values, and return addresses. Fig.1 gives a representation of such a window.



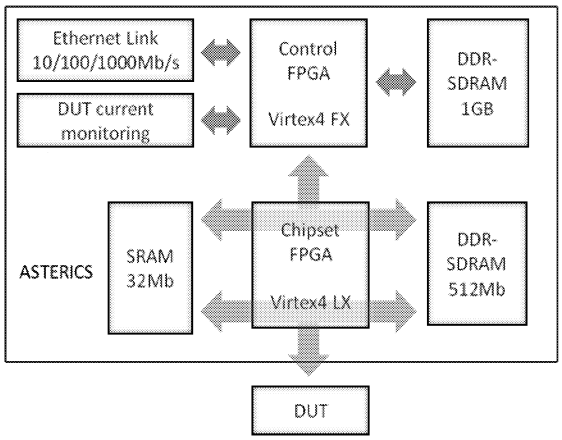
Figures 1: Register window of the LEON processor

Software under test: Gaisler Research provides a portage of the eCos runtime system. This operating system has been designed to support embedded applications with real-time requirements. Preemptability, minimal interrupt latencies and all the necessary synchronization primitives are the real-time main features provided. eCos also provides the environment needed by most of the applications like memory management, exception handling, C and math libraries. In This study, eCos will manage the execution of an AOCS software and a matrix multiplication. The main functionality of the AOCS software is to realize the function of the Altitude Control System of the CNES PRONAOS project. This project aims at the stabilization of a stratospheric balloon embedding a telescope for the observation of the Cold Universe in the sub-millimeter domain. This stabilization is achieved by a control loop based on a Kalman filter depending on the data given by many kind of sensors (inertial motion, GPS, stellar). In this study, the sensor data are fixed and written into the memory of the LEON processor between each execution of the AOCS software.

The matrix multiplication software is a simple program aiming at computing the product of two 8x8 matrices. Such a program is widely used as a benchmark for radiation ground testing and for fault injection experiments.

B. Test Platform

The test platform used for fault injection is an upgraded version of the one presented in (Faure et al. 2002). Fig.2 depicts a block diagram of the test platform. It is built around two FPGAs. The first one, the Control FPGA, manages the communication between the user's computer and the testbed. The second FPGA, the Chipset FPGA, acts as a memory controller. It shares the memory between the DUT (Device Under Test) and the Control FPGA. A control flow checker is also implemented into this FPGA in order to check the correct operation of the DUT. In this study, the DUT (the LEON3 processor) is directly implemented in the second FPGA.



Figures 2: Block diagram of the test platform used for fault injection experiments

C. Test Method

In order to ensure that the LEON3 processor executed correctly its program, the execution flow of the program is split into 2 phases:

- The startup phase where both the processor and eCOS are configured
- The running phase where the processor executes the applications

The duration of the execution of these two phases is measured with counters implemented into the FPGA. If a counter value is different of the one expected, a timeout event is counted.

D. Software development environment

The program executed by the processor was compiled using a GCC based C compiler.

E. Fault injection method

Several methods based on hardware or software strategies can be used for the simulation of SEUs. Software methods to simulate SEU depend on the DUT available description level. Three approaches can be used:

- SPICE level fault injection (Clark et al. 2003)

- VHDL/Verilog level (Massengill et al. 2000)
- Use of a Instruction Set Simulator (Velazco et al. 2002)

Injecting faults at the hardware level for processors can be done by means of:

- A laser beam (Fouillat et al. 2004)
- FPGA: the RTL description of the target circuit is implemented with some modifications allowing the alteration of memory cells (Civera et al. 2001)
- Hardware/software means, using suitable signals (Interrupt, Exception,)

The CEU approach is based on the use of interrupt-like signals available in most of the processors. Indeed, the code associated to the interruption routine can be used to alter a memory cell, randomly selected, among those accessible through the instruction set. The randomness of the SEU's occurrence instants can be easily implemented by random assess of the considered interrupt signal using a counter. The CEU approach is described in detail in (Velaco et al. 2000). An important contribution of this method is related with the prediction of error-rate based on the sensitivity to radiation measured in terms of static crosssection and the results issued from fault injection sessions. The static cross-section, given in equation 1, provides an estimation of the average number of particles needed to provoke an upset.

$$\sigma_{static} = \frac{\#upsets}{\#particules} \qquad (1)$$

In references (Rezgui et al. 2001) (Faure et al. 2005) are shown results of the application of such a strategy to two different processors, while executing benchmark programs, proving that predicted crosssections fit well with measured ones.
The goal of the present work is to apply the CEU strategy to an operating system running two applications in parallel.

III. EXPERIMENTAL RESULTS

Around 140000 bitflips were injected in the LEON3 memory cells accessible by the instruction set. The occurrence instant and the targeted cell are chosen using a pseudo-random generator.

Because of the duration of the program execution (more than 2s), an exhaustive exploration of the fault's timelocation domain is not affordable. Indeed, the number of potential bitflips that could be injected is given by the number of targets times the number of clock cycle needed to execute the program.
As shown in table 1, two types of faulty behaviors at the system level were detected during the fault injection experiments:

- Application errors: the computed data differs from the expected one
- Loss of sequence: the injected fault provokes a timeout

Table 1: Experimental Results

Program	Errors
AOCS	20811
Matrix	948

For 17474 of the 139261 injected bitflips, the system did not reach the end of the execution.

To analyze in detail the consequences of a fault for a given occurrence instant and target, the LEON3 was simulated at the RTL while it executed the studied application. The simulator gives the instruction and its address being executed at any clock cycle. Given that, we can correlate the injected fault parameters (occurrence instant and targeted bit) with the instruction of the program being executed when the fault occurred.

Among the observed system errors provoked by the injected bitflips, must be mentioned the replacement of all the 32 bit words data memory by a binary pattern meaning 0xDEADBEEF in hexadecimal notation. Such a misbehavior can have critical consequence at the system level and was observed for 11 of the total injected bitflips.
As an example, the detailed log obtained for one of this fault is given below:

- Instant of bitflip: 2558797 clock cycles
- Targeted register: %l1
- Targeted bit: 17

Relating to the simulation allows to determine that the targeted register was used as a copy of the stack pointer. It is clear that an access to a misaligned memory address raises an exception of the processor. Apparently, the operating system exception management was designed to fill the user memory with 0xDEADBEEF pattern.
The comparison of the results presented in (Peronnard et al. 2008a) of the AOCS application running in standalone puts in evidence an increase (about 4 times) of the error rate when the AOCS is executed by the operating system.

IV. CONCLUSION AND FUTURE WORK

This paper shows the preliminary steps to assess the evaluation of the soft-error rate of a real time operating system. The approach, socalled CEU (Code Emulating Upsets), based on the activation at random instants of an interrupt signal was used to inject bitflips in accessible memory cells of a LEON3 processor while it executed in parallel two different programs. This preliminary study put in evidence critical errors and an increase of the sensitivity of applications executed by an operating system.

Future work will apply this approach to other operating systems and will confront fault injection results to those issued from radiation ground testing experiments.

V. REFERENCES

Civera P.; L. Macchiarulo; M. Rebaudengo; M. S. Reor-da and M. Violante. 2001. "Exploiting circuit emulation for fast hardness

evaluation". *IEEE Trans. Nucl. Sci.,* Vol. 48, No. 6, 2210-2216 (Dec).

Clark K. A. ; A. A. Ross; H. H. Loomis; T. R. Weatherford; D. J. Fouts; S. P. Buchner and D. McMorrow. 2003. "Modeling single-event effects in a complex digital device*". IEEE Trans. Nucl. Sci.,* Vol. 50, 2069-2080 (Dec).

Faure F.; P. Peronnard and R. Velazco. 2002. "Thesic+: A flexible system for see testing". In *Proc. of RADECS.*

Faure F.; R. Velazco; and P. Peronnard. 2005. "Single event upset like fault injection: a comprehensive framework". *IEEE Trans. Nucl. Sci.,*Vol. 52, 2205-2209 (Dec).

Fouillat P.; V. Pouget; D. Lewis; S. Buchner and D. McMorrow. 2004. "Investigation of single-event transients in fast integrated circuits with a pulsed laser". R. Schrimpf and D. Fleetwood, Theory for Transmission of Vector Quantization Data, World Scientific.

Gaisler research website. http://www.gaisler.com

Koga R.; W. A. Kolasanski; M. T. Marra, and W. A. Hanna. 1985. "Techniques of microprocessor testing and seu-rate prediction". *IEEE Trans. Nucl. Sci.,* Vol. 32, No. 6, 4219–4224 (Dec).

Ma T. and P. Dressendorfer. 1989. "Ionizing Radiation Effects in MOS Devices and Circuits". New York: Wiley, 1989.

Massengill L. W.; A. E. Baranski; D. O. van Nort; J. Meng, and B. L. Bhuva. 2000. "Analysis of single event effects in combinational logicsimulation of the am2901 bitslice processor". *IEEE Trans. Nucl. Sci.,* Vol. 47, 2609-2615 (Dec).

Normand, E. 1966. "Single-event effects in avionics". *IEEE Trans. Nucl. Sci.,* Vol. 43, No. 15,461–474 (Apr).

Peronnard P.; R. VELAZCO; and G. FOUCARD. 2008 a. "Impact of the software optimization on soft error rate: a case study". *DCIS conference,* Grenoble, (Nov).

Peronnard P.; R. Ecoffet; M. Pignol; D. Bellin and R. Velazco. 2008 b. "Predicting the seu error rate through fault injection for a complex microprocessor". *Proceedings of IEEE International Symposium of Industrial Electronics (ISIE),* 2288-2292 (JUN).

Rezgui S.; R. Velazco; R. Ecoffet; S. Rodriguez; and J. Mingo. 2001. "Estimating error rates in processor-based architectures". *IEEE Trans. Nucl. Sci.,* Vol. 48, 1680-1687 (Oct).

SPARC Inc. 1992. "The SPARC Architecture Manual Version 8". Menlo Park, SPARC Internationnal Inc.

Velazco R.; S. Rezgui; and R. Ecoffet. 2000. "Predicting error rate for microprocessor-based digital architectures through c.e.u. (code emulating upsets) injection". *IEEE Trans. Nucl. Sci.,* Vol. 47, 2405-2411 (Dec).

Velazco R.; A. Corominas and P. A. Ferreyra. 2002. "Injecting bit-flips by means of a purely software approach: a case studied". In *Proc. Of Defect and Fault Tolerance in VLSI Systems.*

VI. BIOGRAPHIES

PAUL PERONNARD wa born in French in 1980. He received the M.S. degree in Micro-Electronic (2006) from the Joseph FOURIER University, Grenoble, France. The defense of his PhD entitled "Method and tools for the evaluation of the sensitivity of integrated circuits) is scheduled in October, 2nd. His research are focused on the study of radiation effects on advanced digital circuits like processors.

ESTEBAN CABANILLAS wa born in Cordoba (Argentina) the 4 January 1986th, he performed his engineer studies in the IUA (/Instituto Universitario Aeronautico/). He is a master student in ENIB (Ecole nationale d'Ingenieurs de Brest). He performed in 2008 a research internship at TIMA, where he plans to be a PHD from 2010.

RAOUL VELAZCO was born in Montevideo (Uruguay) in 1952. He received the Diplomas of Engineer, PhD and Dr. ès Sciences from Institut Polytechnique de Grenoble, respectively in 1979, 1982 and 1990. Since 1984 he is a researcher of the CRNS French Research Agency. He leads the ARIS (Architecture Reliable of Integrated Systems) at TIMA Labs. (Grenoble). His researches concern the methodology to asses the sensitivity to the effects of radiation of integrated circuits and systems, the potential solutions to deal with these effects and related experiments in particle accelerator facilities and in scientific satellites.

WASSIM EL FALOU was born in Lebanon in 1975. He received the Diploma in engineering (electrical-electronics) from the Lebanese University, and a M.S in mathematical modelling from Saint Joseph and Lebanese Universities. He received his Ph.D. degree in 2002 from the University of Technology of Troyes, in surface EMG signal processing. He is currently teaching at several universities in Lebanon, including the Lebanese University. His main research interests are embedded systems design and signal processing.

HAISSAM ZIADE was born in Tripoli, Lebanon in 1959. He received the Ph.D. degree in automatic from the Polytechnic Institute of Grenoble, France. He is actually the director of LASTRE Laboratory at Lebanese University, Faculty of Engineering I, Tripoli, Lebanon. His main research interests are microprocessors and programmable logic design.

INDUSTRIAL SIMULATION

A MULTIAGENT SYSTEM FOR PRODUCTION SYNCHRONIZATION IN SME MECHATRONICS SUPPLY CHAIN

Jihene Tounsi
Université de Savoie
SYMME
France
Jihene.tounsi@univ-savoie.fr

Georges Habchi
Université de Savoie
SYMME
France
Georges.habchi@univ-savoie.fr

Julien Boissière
Université de Savoie
LISTIC
France
Julien.boissiere@univ-savoie.fr

KEYWORDS
Small and Medium Enterprises (SME), Supply Chain (SC), Multiagent, Modelling, Integration, Synchronization Protocol

ABSTRACT

We present in this paper the application of multiagent system (MAS) for modelling and simulation of the dynamic structure and behaviour of the supply chain in SMEs mechatronic context in Savoie. First, a short literature overview describes the context and the contribution of multiagent modelling for SMEs integration. In the rest of the paper, the supply chain concepts and their agentification are presented. On the basis of this correspondence, we propose a synchronization protocol for the operational system (production flow and resources).

INTRODUCTION

The growing need for supply chain integration between the various actors is a major strategic challenge and a strong competitive advantage. Indeed, the supply chain is a complex macro-system, first due to the variety of implicated organizational structures and the relationships between them, and secondly due to the strategic decisions it involves. A company's success lies in its ability to integrate managerial processes of the supply chain but also to coordinate with other actors (Drucker, 1998; Lambert and Cooper, 2000).

As such, Small and Medium Enterprises (SMEs) evolve in an unstable and complex economic environment. In order to survive and to be able to support supply chain's requirements and market requests, SMEs have to collaborate together to achieve their goals without losing their autonomy and identity (Julien, 1997; Villarreal Lizarraga *et al.*, 2005). According to Julien (Julien, 1997), to find equilibrium between big companies and SMEs, SMEs must maintain and even improve their competitiveness in an increasingly competitive market. This competitiveness can be achieved by the control of information and processes but also thanks to a network structure. The theory of SMEs network, which is based on collaboration, cooperation and coordination, is a crucial crossroad for its management strategy. The alliances will reduce their vulnerability and this stresses SCM integration in a global supply chain context. This philosophy is explained by Assens (Assens, 1994): "the enterprises network is based on an autonomous and independent SMEs network which interacts in the context of mutual trust relationships to reduce the uncertainty of their environment. During these local interactions, a global organization form emerges in the absence of a central regulatory entity".

In the French region of Savoie, the industrial environment consists of numerous productive SMEs of the mechatronic industry working together in a cluster to achieve a common goal in a complex global supply chain. Some economic researches defined the Savoie industrial context as a region based on sector and geographical concentration of independent SMEs which have complementary activities. This cluster's success is due to interlinking the different production units and developing relationships characterized by flow regulation, learning and coordination. According to this context description and some investigations, we have drawn three main features of the supply chain which integrates SMEs groups and especially mechatronic: (i) complexity due to the number of autonomous actors in SMEs network and the heterogeneity of their relationships (ii) decentralization due to the dispersion of the geographical production sites in the global supply chain (iii) only a local site visibility for SMEs group in the global supply chain. To fill the gap caused by these characteristics, SMEs coordinate their activities with other sites through physical and data flow (Tounsi et al., 2008).
The lack of work in this area and the growing need expressed by professionals lead us to study the structure of the supply chain essentially composed of SMEs mechatronic, and the different mechanisms of integration defining the dynamics behaviour. To fill this gap, we have studied the research advances in industrial and software engineering to find an adequate solution to the studied context. This solution is based on the multiagent approach to model and simulate the static and dynamic aspects. Indeed, the benefits of multiagent systems (MAS) are widely recognized in literature. Agent technology allows new approaches for modelling and simulation of complex distributed systems. Hence, on one hand, autonomous agents can communicate their preferences, deal objectives with other agents or coordinate together to reach their own objectives or some more global objectives (Bussman et al., 2004). On the other hand, the supply chain is composed of actors or entities which evolve in an organization and interact to achieve a collective purpose. This analogy leads to multiagent approach being a framework naturally oriented to model the supply chain.

In this paper, we focus on the description of the dynamic behaviour adopting the multiagent approach. This work is divided into three main parts. The first part discusses on one hand the different mechanisms of management integration in

the supply chain domain and especially in the SMEs mechatronic field; and on another hand, the contribution of multiagent system to implement this dynamic behaviour. In the second part, we present the agentified domain metamodel which describes the static modelling of the context. The third part introduces the behaviour of the identified concepts in the physical system synchronization, before a conclusion is drawn.

LITERATURE OVERVIEW

In this section, we define the different communication mechanisms to integrate the potential actors in the supply chain. We also present the common communication mechanisms in the SMEs cluster and in the global supply chain. Moreover, in the last part we introduce the contribution of the multiagent paradigm to structure the decision-making and the dynamic behaviour of the supply chain actors.

Communication mechanisms in the supply chain

The primary factor of management practices integration in a supply chain is the use of a communication mechanism to synchronize the flows (physical, informational, decisional and financial). In this work, we define a communication mechanism as a "*framework formalizing interaction between different actors in the network according to their managerial relationship characteristics*". In a supply chain, we can find several types of actor's relationships depending on the nature of their goals and their degree of involvement. Indeed, Wong *et al.* (Wong *et al.*, 2004) highlight that in a supply chain, integration is based on different types of relationships that are: *coordination*, *cooperation* and *collaboration*. In many cases, these mechanisms are used with confusion and as being similar. However, according to the literature, each mechanism has its own characteristics. Before describing a relationship between two actors in a supply chain, one must consider the criteria relating to: objectives, decision-making, risks, profits, formal exchanges and sharing product transformation.

According to Mattessich *et al.* (Mattessich *et al.,* 2001), cooperation infers a less formal relationship and involves a lower degree of commitment to join goals. A cooperative relationship entails information sharing as needed, and authority is retained by each organization so that there is virtually no risk. In this kind of communication mechanism, resources and rewarded are separately treated. The product transformation is supported individually by each organization. The second communication mechanism to consider is the coordination which is defined in literature as a process to bring into a common action, movement or condition, or to act together in a smooth concerted way. Compared to cooperation, coordination leads to more formal relationships and an understanding of compatible missions. Planning, division of roles and communication channels are well-defined. Resource allocations are mutually acknowledged, and risks and benefits may be shared or compensated. However, authority may still rest with the individual organization and the product transformation is handled individually. Finally, the third communication mechanism is the collaboration. It is defined as a mutually beneficial and well-defined relationship entered into by two or more organizations to achieve common goals. Collaboration is distinguished from coordination by its collaborative structure which determines a joined authority structure. Resources are pooled or jointly secured, and the product transformation is shared. Table 1 summarizes the different criteria for each communication mechanism.

	Collaboration	**Cooperation**	**Coordination**
Objective	Common	not grouped	Common
Decision-making	Common	Individually	Common + actor decision maker (in some cases)
Benefits	Shared	Individually	Shared
Resources	Shared	Individually	Individually (but known by others)
Formalism	Formal	Few formal	Formal
Risks	Shared	Not shared	Shared
Temporal aspect on product transformation	Shared	Individually	Individually

Table 1. Communication mechanisms (Tounsi *et al.,* 2008)

Integration of SMEs mechatronic network in the supply chain

Previously, we have identified different communication mechanisms that define the management framework of process and information. On the basis of this literature analysis, we studied the different integration mechanisms in the supply chain composed essentially of SMEs mechatronic in Savoie. So, briefly, the global supply chain is divided into many sites geographically dispersed. A cluster of some SMEs mechatronic in Savoie is integrated in this global network but only with a local visibility of the production sites. However, the SMEs cluster is not isolated and it communicates with the other nodes through the different types of flows. The study of the global supply chain and the SMEs cluster identifies two kinds of communication mechanisms according to the criteria defined before. Hence, in the mechatronic cluster, SMEs work together to achieve a common goal. Resources, benefits and products transformation are shared. So, the SMEs collaborate in cluster to reach the local objective. The collaborative process takes place according to the following scenarios:

- *Physical flow synchronization*: in the cluster, the actors will synchronize their information and their actions for production needs. This collaboration is frequent in commonly encountered situations.
- *Local Monitoring*: the monitoring and control of SMEs group will be based on the performance evaluation and the resolution of new perturbations on the physical environment which do not disturb the objectives of other parts of the SC.

Nevertheless, the SMEs cluster will coordinate its activities in order to achieve the global supply chain objectives. At the global level, each node has a limited perimeter of influence. For this reason, a cluster needs to communicate with the other actors to face the environment disturbance by coordinating their activities and local objectives in order to

achieve the global objective. This coordination takes place according to the following scenarios:

- *Initial configuration of the system*: the actors of the network interact in order to set and coordinate their goals in the global supply chain.
- *Environment perturbation affecting all goals*: if a perturbation occurred in a cluster's environment which requires a modification of its local objective and which affects other nodes objectives, the coordination between supply chain sites is necessary in order to find an appropriate solution to the problem.
- *Reconfiguration of the system*: if the actors recognize that the current system configuration cannot achieve the global objective, the interaction between the sites is required to set new objectives according to their experience.

In this paper, we implement the second scenario of the coordination mechanism and the two identified collaboration scenarios.

Multiagent approach for supply chain dynamic modelling

In supply chain modelling and simulation there are two main types of approaches: the equation-based modelling approach and the agent-based modelling one. Parunak *et al*. (Parunak *et al*., 1998) have proven that multiagent systems and agents are more suitable to model the dynamics behaviour of the complex network manufacturing system and to study the impact of flow management between different entities than the equation-based modelling. In fact, agents are more suitable for applications that are decentralized, changeable, ill-structured (dynamic structure) and complex (Parunak, 1998). The multiagent approach provides a framework naturally oriented to model the supply chain. By comparing the supply chain and the multiagent system characteristics, similar concepts and the same organizational practices arise. In fact, both are composed by actors or entities which evolve in an organization and interact to achieve a collective purpose. This analogy leads to multiagent approach being a privileged way to model supply chain systems.

In literature, several research works propose different protocols based on multiagent system for modelling the dynamic behaviour of the supply chain. Most often, these protocols extend the Cooperative Problem Solving model (CPS) proposed by Wooldridge and Jennings (Wooldridge and Jennings, 1999). The CPS defines an abstract way to structure a collective decision-making process in which a group of autonomous agents choose to work together to achieve a common goal. In more details, this model is deployed in four stages:

- *Recognition*: in which an agent identifies the potential for interaction.
- *Team formation*: in which an agent solicits assistance.
- *Plan formation*: in which the newly formed collective attempts to construct an agreed joint plan.
- *Execution*: in which members of the collective play out the roles they have negotiated.

In the following, we describe the implementation of the different collaboration and coordination processes defined previously. These protocols apply the CPS abstract model and structure the interaction between the domain concepts in order to simulate the physical system synchronization.

AGENTIFIED DOMAIN METAMODEL

First, it is necessary to describe the architectural and functional properties of the domain concepts. In this section, we present the agentified domain metamodel which merges the domain concepts with multiagent concepts. The domain metamodel generation and its agentification are based on ArchMDE (Architecture Model Driven Engineering) development process which draws its metamodelling steps in MDE (Model Driven Engineering) approach. MDE (Kent, 2002) promotes the separation and combination of models in order to control the software development in its different phases (from the analysis to the implementation). In fact, in our research context, we identify two metamodels related to ArchMDE approach. The first one describes the functional concepts and properties of SMEs Mechatronic supply chain. The second one defines a multiagent metamodel. A combination of the two metamodels will generate an agentified domain metamodel (see figure 1). Only the main results of this approach are outlined in this section. More details can be found in (Tounsi *et al.* 2009a).

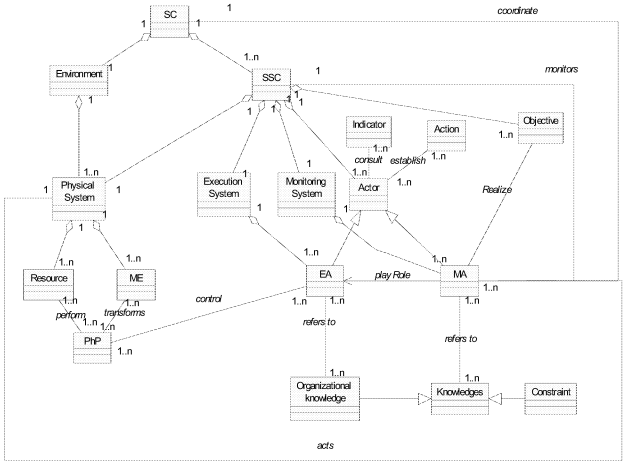


Figure 1 Domain metamodel (top-down)

The domain metamodel is an UML class diagram which represents the concepts defining the architecture and the properties of the global supply chain and the SMEs group (Tounsi *et al.,* 2009b).

- *Supply chain (SC)*: this concept defines the root of the domain modelling. All the other concepts will detail this one.
- *Environment*: this concept models the space allocated to the product flow and management through the internal resources as well as the external elements able to influence the activities of the supply chain.
- *Sub Supply Chain (SSC)*: the SSC represents a group of SMEs which collaborate to achieve an internal aim and/or the overall objective of the supply chain. The SSC is

responsible for the management of the product flow in a certain stage of its life cycle. The SSC is composed of 3 layers involving some particular concepts and playing a specific role.

These 3 first concepts define the abstract framework of the context. Thus, according to the domain metamodel a supply chain (SC) is composed of one or more organizations (SSC) and a physical environment. SSC at this level represents a SMEs group. A Detailed presentation of the internal architecture of the SSC and its perimeter of influence (visible part of the environment to the SSC) follows. These concepts represent the second level of the domain metamodel.

- The *Monitoring System* models the intelligent layer of the SSC. It controls and monitors the two other layers through the information provided by the Execution System. The *Monitoring Actors (MA)* are the main elements of this layer. They model the intelligent actors of SSC and establish metrics to evaluate the performance of the group and consequently act on the two other layers. Therefore, the MAs are the components responsible for the control and decision-making in SSC but also for the coordination activity in the global supply chain.
- The *Execution System* is the reactive layer of the SSC. It deals with two main roles: (i) it ensures the synchronization of the physical flow according to the information gathered from the *Physical System*, (ii) it observes and corrects the *Physical System* if a perturbation occurs. In abnormal situations, the *Execution System* refers to the *Monitoring System* for coordination and decision-making. The *Executive Actors (EA)* are the principal entities of this layer. The EAs mainly model the reactive actors and occasionally MAs having reactive behaviour in this layer.
- The *Physical System* is the visible part of the environment from the SSC. It corresponds to the influence perimeter of the SSC. This layer is composed of passive elements controlled by the two other layers of the SSC. Two main concepts are identified: the *Moving Entity (ME)* modelling the product in circulation and the *Resource* modelling the production means.

Finally, the following concepts considered as data support, allow some other concepts to play their roles in the metamodel:

- The *Physical Process* (PhP) describes the sequences of processing stages of the product. The PhP is a concept to be integrated within a domain metamodel in order to define the tasks that can be handled by the *Execution System*.
- The *Indicator Base* represents a database recording the indicator measures. The EA detects a Physical System deviation according to the gathered information within this database.
- The *Action Base* represents a database that stores the actions to apply when facing an indicator deviation.
- The *Objective* models the global goals of the supply chain or of the SSC ones.
- The *Knowledge Base* represents a database including all knowledge needed by the actors to make the right decision. A n element of the Knowledge Base can be an organizational knowledge or a constraint.

At this step, the domain metamodel describes the supply chain in SMEs mechatronic context. This metamodel is merged with a multiagent metamodel in order to obtain a correspondence between each domain concept and multiagent one. The agentified domain concept implements the structural properties and dynamic behaviour of the multiagent one. Figure 2 draws the multiagent metamodel according to the "vowel approach" introduced by Demazeau (Demazeau, 1996).

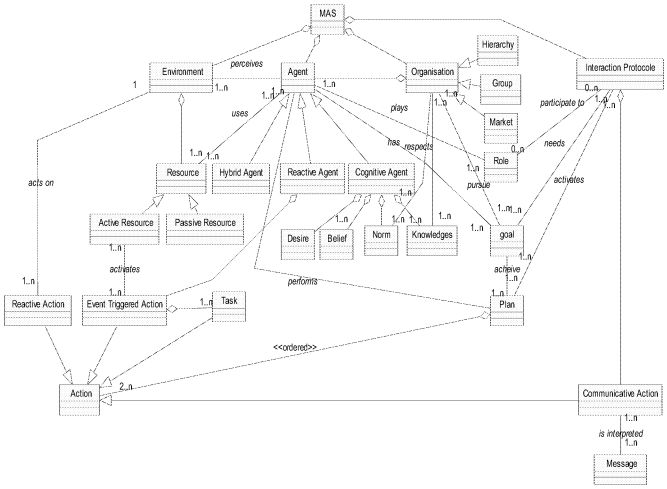


Figure 2 Multiagent metamodel (Tounsi et al., 2009a)

So, according to these two metamodels, the following table highlights the correspondence between the domain concepts and the multiagent ones.

Domain concepts	***Multiagent concepts***	***Description***
Supply Chain (SC)	MAS	By analogy, the root of the domain metamodel corresponds to the root of the multiagent system.
Environment	Environment	In the two metamodels, environment is the space defining all things external to agents and necessary to manage the SC.
Sub Supply Chain (SSC)	Organization	It is an organization composed of two groups of agents.
Physical System	Resource	It is all needed resources for one agent or a group of agents to manage the group (perimeter of influence).
Resource	Passive Resource	It is an allocated resource to the agent to perform its task.
Moving Entity (ME)	Active Resource	It is the product in circulation. ME activates the behaviour of the reactive agents.
Physical Process (PhP)	Task	It is a task or a physical activity to be handled by reactive agents.

Monitoring System	Group	It is a group of cognitive agents which collaborate in the SSC and also coordinate the activity of the organization with other organizations.
Execution System	Group	It is a group of reactive agents which collaborate in the SSC.
Actor	Agent	An actor can be a cognitive agent or a reactive agent according to its decisional granularity.
Executive Actor (EA)	Reactive Agent	EA perceives physical system and acts on it according to the observation.
Monitoring Actor (MA)	Cognitive Agent	According to the collected information and the historic of situation and action, MAs group monitors the SSC to reach a goal and achieve its activity.
Objective	Goal	A SSC have a goal to reach. This goal is coordinated with other nodes' goalss.
Indicator	Belief	The agents act on the environment according to the indicator measures. However, a MA monitors the SSC according to the historic of these measures.
Action	Plan	It is an action or a plan to apply when facing a perturbation.
Knowledge	Knowledge	It is all needed knowledge to the agents to act in appropriate way.
Organizational knowledge	Knowledge	Each agent owns a list that contains the information about other agents from the same SSC or the global SC. This list stores knowledge about the name of the agent, the task that it performs and its resources.
Constraint	Knowledge	The MAs take decision according to their objectives and their beliefs. In the same time, there's some constraints (about product, or other SC where the group involves) that MAs group must take into account in decision-making.

Table 1. Correspondence domain concepts and multiagent ones

PHYSICAL SYSTEM SYNCHRONIZATION

The SSC is responsible for the synchronization of the involved Physical System to achieve its task. This activity consists in applying a communication protocol according to the nature of the interaction framework. In this section, we describe the collaboration and coordination processes to implement in the Execution System and Monitoring System in order to synchronize the Physical System. These protocols are built according to the CPS abstract model.

In accordance with the agentified domain metamodel, the Execution system is responsible for the synchronization of the physical process (PhP) in common situations. Indeed, Executive Actors (EAs) which are reactive agents, synchronize PhP by taking into account the availability of resources. Then, on the basis of the agentified domain concepts, the synchronization protocol in the Executive System is shown in Figure 5:

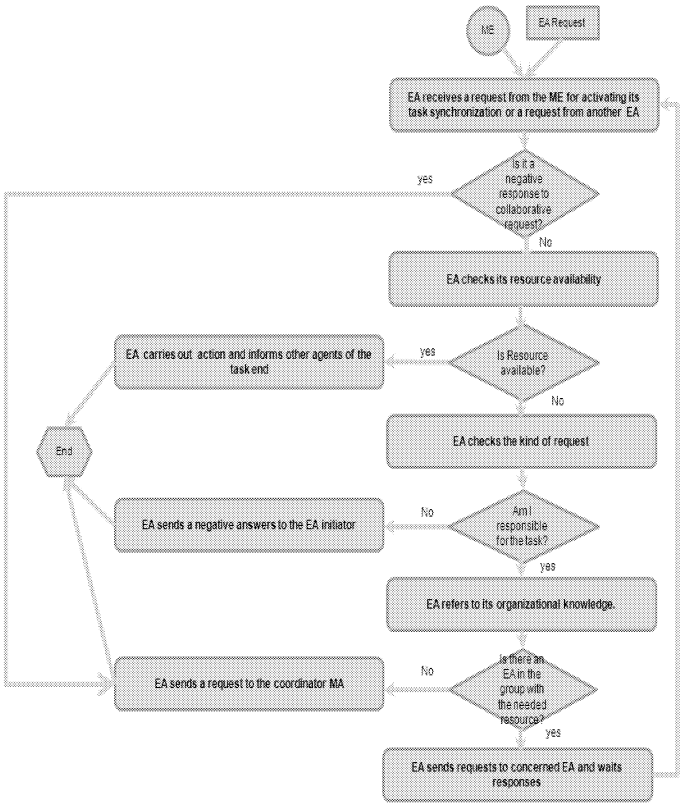


Figure 3. EA Synchronization Behaviour (Tounsi et al., 2009b)

- An Executive Agent (EA) receives a request and reacts according to its type. Three kinds of request can be distinguished: a ME request, a collaborative request from another EA or a negative response to a collaborative action initiated by the agent itself.
- If the request is a negative response to a collaborative demand that the EA initiated: the EA sends a request to the coordinator agent of the monitoring system.
- If the request is a synchronization need coming from the Moving Entity (ME) or a collaborative request from

another initiator agent, EA checks the availability of the concerned resources.

- If the resource is available, the EA carries out its task, updates the state of ME and informs the other agents from the executive system and the coordinator agent of the end of action.
- If the resource is unavailable and the EA has been solicited by another executive agent to achieve the task, it sends a failure request to the initiator.
- If the resource is unavailable and the EA is in charge of the task then it seeks in its organizational knowledge an agent from the Execution System of the SSC that might have the needed resource.
- If the agent finds into its organizational knowledge an agent that can handle the task, it delegates the responsibility of the task. In this case, the collaboration process of the concerned agent will be activated and follows the same sequence.
- If the agent does not find another agent having the needed resource to handle the task, it sends a request to the coordinator agent. This agent is a monitoring agent (MA) that receives requests from the Execution System. MA sends the information to other monitoring agents of the SSC in order to find a solution.

However, in unusual situations, the Executive System refers to the Monitoring System. In this case, MAs group evaluates the situation according to the defined objective and establishes an actions plan. If the objective is not reached, the MA needs to consult other SSCs to find a suitable solution. So, the synchronization protocol in the Monitoring System can be described according to the follow steps:

- In the Monitoring System, a Monitoring Actor (MA) has the responsibility to check all the received requests and diffuses them to other MAs in the layer. Three kinds of requests can be distinguished: EA request, answer to a help request or a help request from another SSC in the global supply chain.
- If a MA coordinator receives an EA request then it diffuses the information to other MAs. In this case, according to the objective, the group evaluates the situation. Two cases may arise: the problem has no impact on satisfying the SSC objective or the objective is deviated.
- If there is no impact on the objective, the MA tries to find an internal solution according to its desire, belief and constraints. If a solution can be found, the MA coordinator spreads the actions plan to the Executive System.
- If the objective is deviated or an internal solution can not be reached. The MA group sends a Help Request to other SSCs through the MA coordinator and waits responses.
- If all responses are received, the MA coordinator classifies them by a reception date of help. The selection will be diffused in the Monitoring System. According to their beliefs, desires and constraints, MAs group chooses the suitable answer and diffuses the action plan to the Execution System. In this case, the EA updates the ME state.
- If the request is a Help Request from another SSC, the MA coordinator spreads the request to the Monitoring System. In this case, the Monitoring System evaluates the demands according to internal criteria (Objective, Constraint, Belief, and Desire). If the SSC can provide assistance, it makes an offer to the SSC initiator or it sends a negative response.
- SSC initiator chooses the suitable offer and sends a confirmation to the selected SSC and a cancellation response to other bids.

The following figure shows the sequence of messages between the SSCs. This diagram represents the coordination process in the global supply chain in order to synchronize the physical flow in the case of a perturbation case (SSC cannot reach the internal aim).

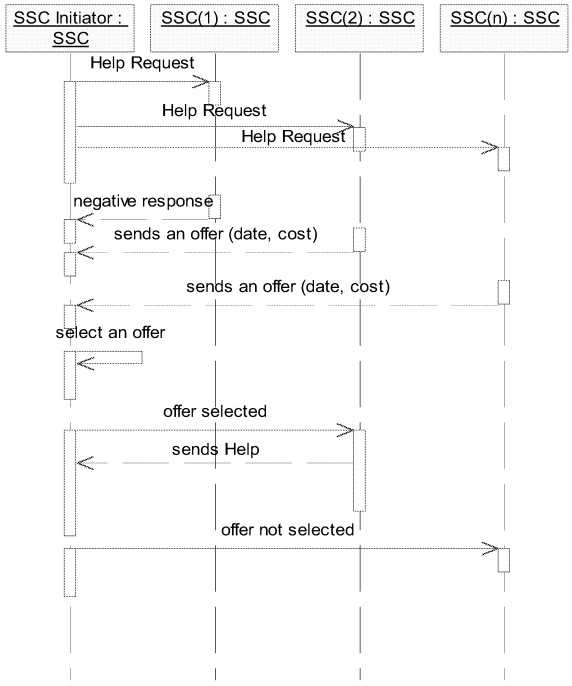


Figure 4. SSC coordination process

CONCLUSION AND PERSPECTIVES

The present research work focuses on the integration of mechatronic SMEs into the global supply chain. These SMEs clustered in group are localised in Savoie, a French region. Communication protocols for physical environment synchronization are proposed. These protocols describe the dynamic behaviour of agentified domain concepts, since the modelling solution is based on multiagent paradigm. The multiagent system defines the behaviour characteristics of the domain concepts and consequently the establishment of communication protocols. Thereby, this article introduces a stage from a global solution. Now, our research work focuses on the implementation of the performance evaluation processes in the SMEs group and in the global supply chain. All the concepts and processes will be encoded within a simulation platform.

THANKS

We thank the APS (Assemblée des Pays de Savoie) and the Laboratory SYMME (Systèmes et Matériaux pour la Mécatronique) who funded this work through a research grant.

BIBLIOGRAPHY

Assens C., 1994. Du modèle bureaucratique au modèle organique, Research Notebook n°232, DMSP.

Bussmann S., Jennings N.R, Wooldridge M., 2004. *Multiagent Systems for Manufacturing Control: A Design Methodology*, Springer Series on Agent Technology, Springer.

Demazeau Y., 1996. "*Vowels*", Invited lecture, *IWDAIMAS96.*

Drucker P.F, 1998. *Management's new paradigms.* Forbes, October, pp. 152-177.

Julien P.A., 1997. *Les PME bilan et perspectives*. 2e edition, Economica, Paris, France.

Kent S., 2002. *Model-driven Engineering*, *IFM 2002*, vol. 2335 of LNCS, Springer-Verlag, pp. 286-298.

La Londe B.J and Masters J.M, 1994. *Emerging logistics strategies: Blueprint for the next century*. International Journal of Physical distribution and Logistics Management. vol 24, n°7, pp. 35-47.

Lambert D.M. and Cooper M.C., 2000. *Issues in supply chain management.* Industrial Marketing Management, 29, n° 1, pp. 65-83.

Mattessich P.W., Muray-Close M., and Monsey B.R, 2001. *Collaboration: what makes it works*. 2 edition, Amherst H. Wilder Foundation.

Parunak H., Savit R., Riolo R.L, 1998. *Agent-based modelling VS equation-based modelling: a case study and user's guide.* Proceedings of Multi-agent systems and agent-based simulation (MABS'98), Springer, LNAI 1534.

Parunak H., 1998. *Wat can agents do in industry, and why? An overview of industrially-oriented R&D at CEC.* Second international workshop on cooperative information agents, CIA'98, M.Klusch.

Tounsi J., Boissière J., Habchi G., 2008. *A conceptual model for SME Mechatronics supply chain.* 6th International Industrial Simulation Conference (ISC'08), Lyon, France, pp. 273-280.

Tounsi J., Azaiez S., Habchi G., Boissière J., 2009a. *A multiagent approach for modelling SMEs mechatronic supply chains*. 13th IFAC Symposium on Information Control Problems in Manufacturing, Moscow, June 3-5.

Tounsi J., Boissière J., Habchi G., 2009b. *Multiagent decision making for SME supply chain simulation*. 23rd European Conference on Modelling and Simulation (ECMS), Madrid, June 9-12, pp:203-211.

Villarreal Lizarraga, C.L., Dupont L., Gourg D., Pingaud H., 2005. *Contributing to management of shared projects in SMEs manufacturing clusters.* 18th International Conference on Production Research (ICPR-18), Salerno, Italie.

Wong C.Y., Johansen J. and Hvolby H.H., 2004. *Supply chain coordination Problems: Litterature Review from organization, economic and operations perspectives*. Working Paper no. 08-04, Center for Industrial Production, Aalborg University, pp. 1-23

Wooldridge M. and Jennings N.R 1999. *The cooperative Problem-Solving Process*. Journal of Logic Computational., Vol. 9, pp. 563-592.

SIMULATION OF DYNAMIC ALLOCATION AND ROUTING IN FMS USING POTENTIAL FIELD

Nadine Zbib
Yves Sallez
Damien Trentesaux
UVHC, LAMIH, CNRS – UMR 8530
Univ. Lille Nord de France
F-59313 Valenciennes, France
E-mail: Nadine.Zbib@univ-valenciennes.fr

KEYWORDS
Manufacturing, decision making, discrete simulation, resource management, dynamic system.

ABSTRACT

The aim of this paper is to point out the capabilities of potential fields for the dynamic control of flexible manufacturing systems. Products sense the fields emitted by resources according to the service they request. Resources adapt their emitted field according to their states and their workloads. Routing and allocation are then simultaneously solved dynamically. Simulations are made using Netlogo software. To illustrate the features of our model, we compare the results of our application to a concurrent dynamic allocation approach based upon the contract net protocol.

INTRODUCTION

The manufacturing industry must continually deals with the increasing complexity of product requirements, as well as with the desire to decrease costs and increase control system adaptability to environmental changes. In recent decades, scientific development in the field of production has provided greater equipment automation and defined new architectures, including the heterarchical/non-hierarchical architectures that play a prominent role in Flexible Manufacturing System (FMS), leading to entities (products & resources) that are more independent, adaptative and reactive.

Traditional approach usually leads to centralized or hierarchical control structures. Due to the complexity of manufacturing problems, the usual practice has been to split the overall problem into hierarchically-dependent functions that operate within decreasing time-ranges, such as planning, scheduling and/or monitoring. This traditional approach can hardly manage unpredictable events (e.g., machine breakdowns, urgent order) and mass customization (large amount of products, each of them being particularized to each customer). More adaptable and agile control must be designed.

In this context, the objective of our work is to show how the concept of potential field can provide very reactive and agile control of production systems.

The first section describes the different approaches that are usually proposed to control interactions among entities. The second section describes a modeling approach for potential fields. Then, a potential field model dedicated to production control is proposed. Finally, a simulation and a validation are presented on a flexible assembly cell.

STATE OF THE ART

Potential fields concern the interactions among several entities in a distributed way (in opposition to a centralized way). In distributed system, many interactions are used to ensure effective decisions by entities. The distributed architectures can be categorized in four types: bionic & bio-inspired, as proposed by (Okino 1993 or Dorigo and Stützle 2004), multi-agent, as proposed by (Maione and Naso 2003), holonic, as proposed by (Van Brussel et al. 1998), and heterarchical, as proposed by (Trentesaux et al. 1998). An analysis of the state-of-the-art has been recently published by (Trentesaux 2007).

The vast majority of agents and holonic models are based on the contract-net protocol (Smith 1980). The contract net protocol which is a widely-accepted problem solving model in distributed Artificial Intelligence presents the benefit of being general, intuitive and easily applicable to different contexts. This protocol has been developed in order to specify the resolution of problems by a group of entities. The distribution of tasks is made by using a negotiation process: a discussion done between the entities relating the tasks that should be executed and those which are capable to execute the candidate task ends in electing an entity that handles the task.

In the bionic or bio-inspired approaches, stigmergy is often used; see for example ACO (ant colony optimization) algorithms (Dorigo and Stützle 2004). Another example is that of the potential field (Ferber

1995), but this technique, applied widely for example in robotics has rarely been used in the context of production (Berger et al., 2009).

Our objective is to test and implement this concept in the context of production system. Historically in our research laboratory, the interactions among entities (resources only) were governed by the *contract-net*. More recently, we were interested to *stigmergy* for routing products (Sallez et al. 2009). The application of this third interaction mechanism (*potential field*) will give us a fairly comprehensive view on the advantages and disadvantages of each concept to propose a multi-model approach.

POTENTIAL FIELD APPROACH

The concept of potential field is used in reactive systems to determine the behavior of entities. It is relatively a new technique that owns its roots in a radical critique of conventional planning techniques to determine a way forward. In the field of navigation of robots, searchers used this concept in the planning of movements, thanks to their ability to act in areas of continuous real time (Barraquand and Latombe 1989, Khatib, 1985 and Ferber 1995 and Mamei et al., 2006). Two types of potential fields are designed: a repulsive potential field to obstacles and an attractive potential field to the desired destination. A robot can then follow the path that has the greatest attractive potential field while avoiding obstacles.

This approach is used to solve the problem of positioning in a continuous environment. Moujahed (Moujahed et al., 2007), applied this approach for a better position to the bus stops on existing lines.

In manufacturing systems, (Ueda et al, 1998) have also used potential fields to ensure a dynamic allocation of products among a set of resources. They have applied this concept to a system line-less production where products (carried on AGVs) are treated by robots on mobile platforms. The field is the superposition of an attraction and a repulsion field: each machine generates an attraction force to obtain products. Each product senses the cumulated attraction field of the machines and moves accordingly.

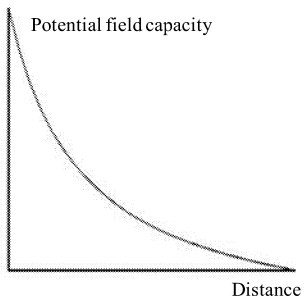


Figure 1: Variation of potential field

The capacity of potential field is inversely proportional to the Euclidean distance between the entity to attract and the center of attraction. The more the distance is small; the more the attraction is important (cf. Figure 1).

In all these potential field models, entities emit attractive potential fields propagating in a plane in all directions. Mobile entities move in 2D within a field, in any direction.

These characteristics are not relevant to typical production system and we must adapt the notion of distance according to the application context.

Figure 2 shows this type of potential field. 1D travels are rarely considered (which means that entities must move within a fixed routing network, such as pallets in a transport system, or vehicles in a city). In this paper:

- Only resources (Rj) emit fields, while products (Pi) sense and follow the gradient of these fields according to the services they request.
- Only attractive fields are considered, not repulsive ones.
- Potential fields are designed to take into consideration the workload of resources and the service they provide.

In future studies, more complex potential fields will be evaluated (e.g., both products & resources emit attractive & repulsive fields). The following section presents then our model.

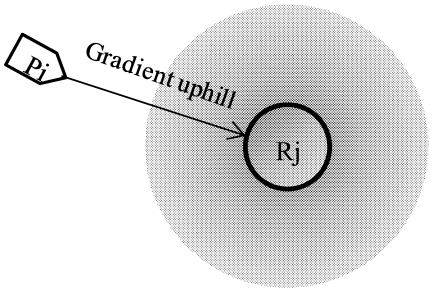


Figure 2: Product Pi moving to the resource Rj

PROPOSITION

Modeling approach

In our model, we consider active entities, which can be a resource or a product. An active entity is an entity that is able to inform, communicate, decide and act in order to reach its goals. For more detail about the concept of *active entity*, see (Zbib et al. 2007). Compared to, for example, contract net, which can be only used to dynamic allocation functionality (e.g., tasks to resources), potential field naturally enable dynamic allocation functionality simultaneously to dynamic routing functionality.

Figure 3 shows a general view of allocation and routing processes. In this figure, three resources are shown, each resource being able to satisfy services: R1 (S1, S2), R2 (S2, S3) and R3 (S1, S3). In this figure, product P1 needs to satisfy services S1, S2, S3 sequentially. More globally, this defines a flexible manufacturing system (FMS) which is the considered application case in this paper.

The topology of an FMS can usually be represented using graph theory. Resources, located on nodes, emit potential fields and active products follow a succession of arcs from node to node to complete its manufacturing. The moving of product is constrained by the topology of the FMS.

Other criteria other than the distance between product-resource should be taken into account to evaluate the intensity of the fields. In each resource, the capacity of the queue, the time of executing a service, the state of the resource (idle, busy, failed), the quality of service provided are to be considered.

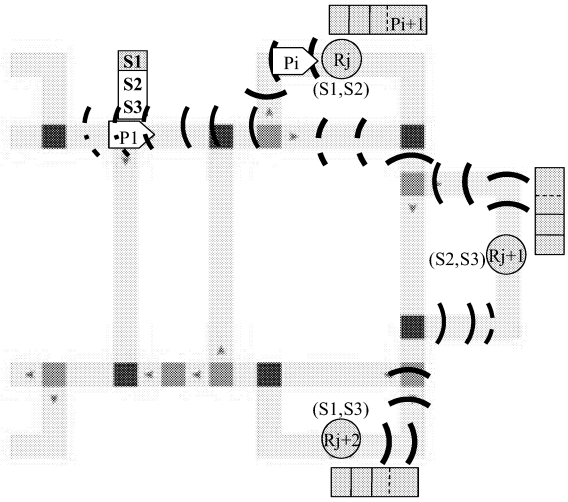


Figure 3: model of our graph

Notations

Notations are:
Let R = (Rj) the set of resources located on nodes, and P = (Pi) the set of active products that move in a manufacturing system to get a list of requested services S = (Sk). Each Rj diffuses an attractive potential field to attract active product Pi and provide one service Sk.

A product Pi, seeking to satisfy a service S_k, goes to the resource emitting the more intense potential field for S_k.

The potential field is emitted in 1D according the topology of the graph (cf. Figure 3) and its value decreases by moving away from the resource.

In this context, the distance between two points is not the Euclidean distance but is the sum of the lengths of arcs joining these two points. Thus the example in Figure 4, the distance between the starting node A and destination node E, is expressed as dist (A, E) = d1 + d2 + d3 + d4, di is the length of each arc i traveled.

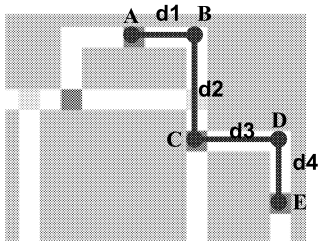


Figure 4: formula of distance

To each resource a queue is defined, $Ta(P_i, R_j)$ denotes the waiting time of a product P_i in the queue of a resource R_j.

$Tm(P_i, R_j)$ denotes the movement time of a product Pi to the resource Rj. V_{pi} is the speed of P_i during its movement.

The execution time for a service S_k to a product P_i in the resource R_j is $Ts(R_j, P_i, S_k)$. Total processing time T for a product P_i, is the time a product requires his service S_k on R_j. The form of T is therefore expressed in the formula:

$$T(R_j, P_i, S_k) = Ta(P_i, R_j) + Ts(R_j, P_i, S_k)$$

Potential field intensity is then defined as:

$$Cattr(R_j, P_i, S_k)_t = Cattr(R_j, P_i, S_k)_0 * \left(\frac{1}{1 + (V_{pi} * Tm_{Pi,Rj})} * \frac{1}{1 + T(R_j, P_i, S_k)} \right)$$

This formula takes into consideration the velocity of products, movement times, number of products in the queue of a resource and the total processing time.

The initial value of the intensity of the potential field is $Cattr(R_j, P_i, S_k)_0$. At time 0 this value is the maximal capacity of potential field of each resource, when none product is in the system and all resources are idle. After that, at time t, the availability of resource will decrease when it becomes busy delivering a service S_k to P_i (and / or) when some products are waiting in its queue. This value decreases also if the product has an important movement time, i.e. the distance between P_i and R_j is high.

Active product decisional behaviour

The state graph of Figure 5 describes the successive states of a product.

A product must obtain all services from its list in order to be considered finished. First, the product is moving to reach a resource that gives him a requested service. The product is on the way, it is blocked by an event that disrupts its displacement, or it reaches its resource destination.

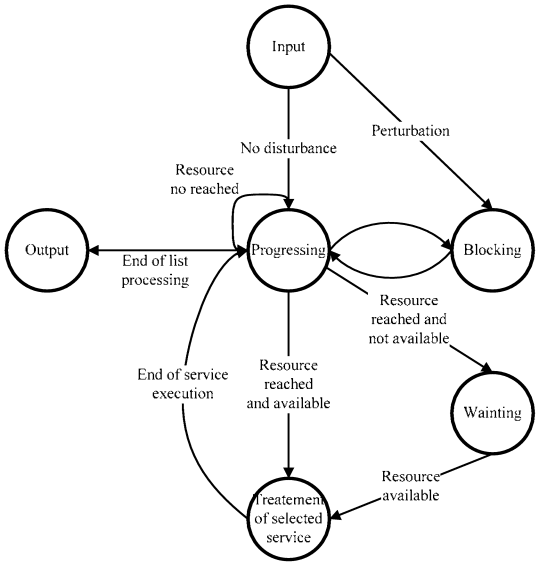


Figure 5: Product states

When the resource is reached and available, the product will be processed, but if resource is not available the product waits for the treatment; possibly, several products are in the queue of the resource, the product remains there waiting until it is selected. When the product is selected by the resource, it is processed. Then, once the list of services is completed, the product leaves the system; otherwise it continues its progress towards other resources.

SIMULATION

The proposed model is naturally distributed, meaning there is no central memorization and processing system. This property influenced our choice of an agent-based parallel modeling and simulation environment. With NetLogo (Wilensky 1999), each modeled entity can be described as an independent agent interacting with its environment. All agents operate in parallel on a grid of patches (i.e., a cellular world), and each agent can read and modify some of the attributes linked to the patches in its proximity. The behavioral rules defined for the agents make it possible to describe agent-environment interaction, which is very important when simulating the potential field concept. Netlogo was chosen because it supports easily all our desired functionalities and provides an intuitive and well-documented programming language with an elegant graphical interface.

Netlogo is based upon two major concepts: turtles (moving entity able to inform and communicate with its environment) and patches (fixed locations on which turtles moves). In our simulation, a product is a turtle moving in his environment, and a resource is a fixed turtle able to emit potential field.

CASE STUDY AND VALIDATION

An existing benchmark developed in our laboratory has been used purpose (Trentesaux et al, 2000). This benchmark used multi-agent simulation where products and resources negotiate using the contract-net protocol to optimize dynamically allocation of tasks. Routing was simplified (not considered or roughly modeled as a constant time delay). We note that in this approach, resources are the only decisional entities for the treatment of tasks. We used the same data to compare our approach to this benchmark.

Experimental context

This case study concerns a FMS system inspired from the flexible assembly cell at the Valenciennes AIP-Primeca pole. This cell is composed of workstations placed around a flexible conveyor system based on Montrac technology (Montech 2008). This system is a monorail transport system using self-propelled shuttles to transport products. Topological graph is composed of three types of nodes (cf. Figure 6):

- Service or resource nodes: resources where products obtain services (R1, R2, R3, R4, R5, and R6).
- 11 decisional nodes, where a routing choice between two adjacent arcs must be made (gray node): when a product arrives on a decisional node, it checks the potential field value of neighbours patches to know the direction it must take to reach the best resource.
- 11 convergent nodes (black node).

A product, represented as an arrow (gray), enters and leaves on "I" node of the system.

We implemented in our simulation two types of agents with for each one private variables and parameters: resources – as fixed agents –and products – as a mobile agents –. In this simulation potential field values can be sensed by the products everywhere in the system.

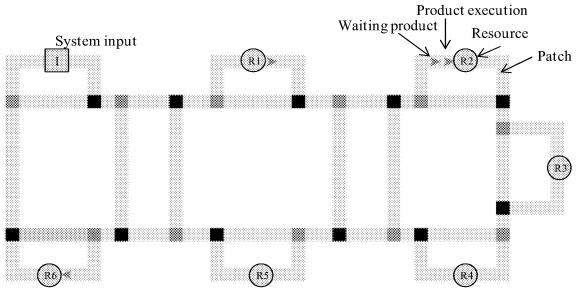


Figure 6: Topology of the FMS

The variables and parameters for a resource agent are given in Figure 7.

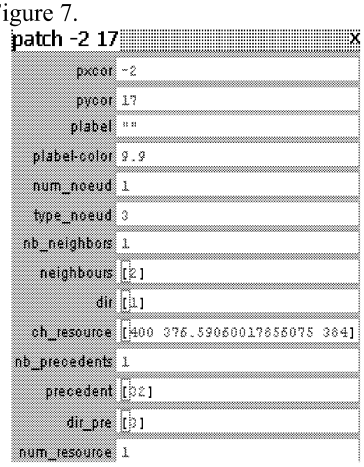


Figure 7: Variables and parameters for a fixed entity (resource agent)

For each resource, the main parameters and variables are: its location (*pxcor, pycor*), the reference number of the node on which the resource is located (*num_noeud*), the node type (1 = convergent node, 2 = decision node, 3 = service node), the number of node neighbors (*nb_neighbors*), the direction to be taken to reach the neighboring node *(dir*), the potential field value (*ch-resource*),this list contains values of potential field for each type of service. Another variable is the precedent nodes, and finally we can show the identifier of the resource (*num_resource).*

The variables and parameters for a product are shown Figure 8. A product with its identifier (*prod)* enters the system with a list of ordered services (*liste)* [S1, S2, S3] to be satisfied, and a list for execution time (*temps_trait)* [T1, T2, T3] for each service. Each P_i selects the first service S1 (*demande)*, and moves to find the "best" resource (according to the potential flow values). When it arrives to a gray node, it checks the neighbours patch to choose the highest value of potential field corresponding to the required service. Product uphilles the gradient to the chosen patch, and moves then, following the light grey patches (cf. figure 6), to reach the targeted resource/service node.

On this node, the product waits during a time *temps* to have *(demande).* When its execution is achieved (*finish_time)*, it selects the next service and moves again. When *prod* is achieved, (*traitement_fini)* is set to true, and *prod* goes out of the system.

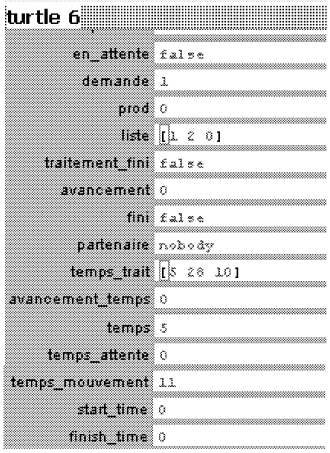


Figure 8: Variables and parameters for a mobile entity (product agent)

In the case study used, flexible capabilities are the following: Resource #1 is able to execute service task S1, resource #2 and resource #3 are able to execute service task S2 and resources #4, #5 and #6 are able to execute service task S3. Each manufacturing order (a set of ordered services to complete and their durations) is composed of three tasks/services, where each task is one among the three types.

Table 1: Data of simulation

Manufacturing orders	*List*	*Processing times*		
1	1-2-3	5	28	10
2	2-3-1	20	11	12
3	3-1-2	18	11	14
4	1-2-3	9	19	23
5	3-1-2	18	10	14
6	2-3-1	15	26	15
7	2-1-3	29	10	10
8	1-3-2	10	15	18
9	2-1-3	14	22	29
10	3-2-1	27	17	9
11	1-2-3	25	11	30
12	2-1-3	14	7	35
13	1-3-2	18	38	23
14	1-2-3	6	20	38
15	2-3-1	18	43	5
16	3-2-1	35	21	8
17	1-3-2	7	35	16
18	2-3-1	12	30	10
19	3-2-1	37	22	13
20	2-3-1	18	32	6
21	1-2-3	7	15	37
22	2-1-3	18	15	32
23	1-2-3	9	12	32
24	3-2-1	27	13	8
25	1-3-2	9	13	28
26	1-2-3	8	17	21
27	3-2-1	25	10	9
28	2-1-3	19	13	21
29	1-2-3	14	31	21
30	3-1-2	26	11	13

30 manufacturing orders have been defined. The list of the 30 manufacturing order is given in table 1.

Results

Scenario 1: optimality

The comparison indicator is the classical makespan value that is the final date, when all tasks have been done.

We propose 6 different simulations according to the growing number of manufacturing orders to schedule: Simulation #1 (5 manufacturing orders), Simulation #2 (10 manufacturing orders), Simulation #3 (15 manufacturing orders), Simulation #4 (20 manufacturing orders), Simulation #5 (25 manufacturing orders), Simulation #6 (30 manufacturing orders).

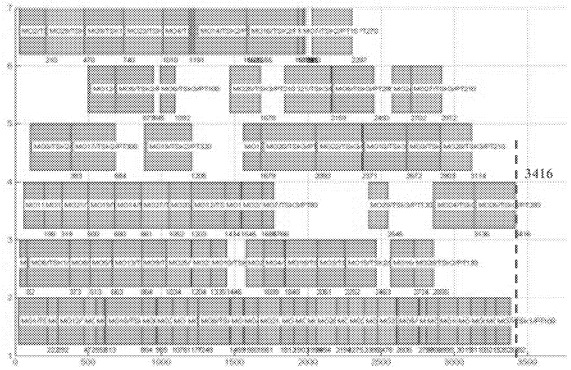


Figure 9: Gantt for simulation #6 (scenario 1)

Figure 9 shows an example of the obtained Gantt for simulation #6, starting and finished times for each product is provided (time is measured by Time Unit). We note that, the resource 1 is overloaded, since it is the only resource that handles the service S1, the treatment of service S2 is correctly balanced between the resource 2 and 3, while the S3 is correctly balanced among R4, R5 and R6.

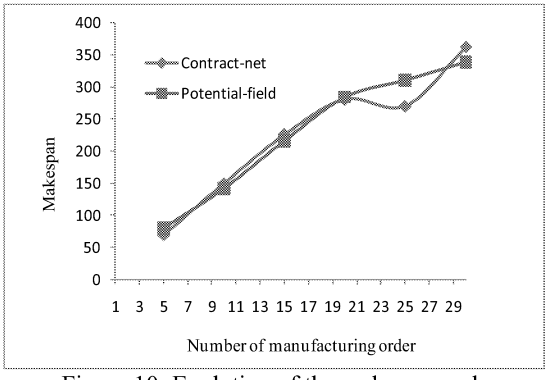


Figure 10: Evolution of the makespan values

Makespan results are given in Figure 10. Most of the makespan values provided by potential field concept are better than those provided by the contract-net based control.

In our results only product is a decisional entity that makes decisions of routing according the potential field value. We show that the execution time increases gradually with increasing number of manufacturing orders, which is logical with the concept of potential field
The decrease of execution time with contract-net in the simulation from #23 to #27 according to the authors depends on the following reason: "*The local and dynamic decisions implying a restriction of visibility induced by a distributed- based approach*".

Scenario 2: reactivity

Reactivity can easily be evaluated by for example, allowing or forbidding replies of resources: when a resource is unavailable (e.g., under maintenance), it will no longer reply to any request. Figure 11 illustrates a situation where resource #2 is available at date 100 and resource #4, at date 200. This implies logically an increase of the makespan value (354). This makespan is also increased with contract-net protocol (407).

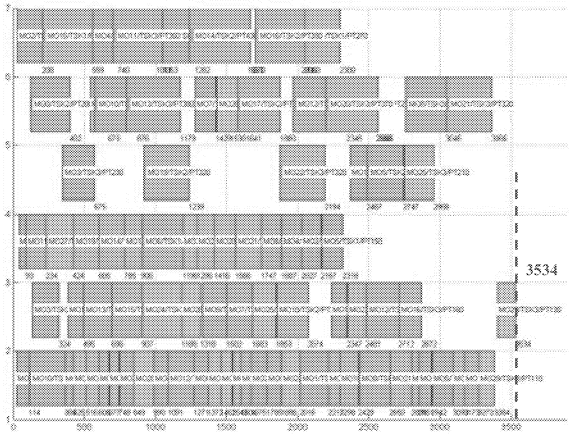


Figure 11: Impact of availability of resources

Scenario 3: scalable & realistic behavior

We extend the complexity of the study by introducing a greater number of products (60 products), to study the ability of the simulation to be scalable. We faced no issue in this case (cp. Figure 12). The number of entities does not imply complex exponential interactions, in contrast to approaches such as contract-net.

Other studies not reported in this paper show the capability of our simulation model to take into account other realistic assumptions often not considered in the literature, such as limited capacity of production stocks, jamming of the routing system, re-routing of products due to quality problems, etc.

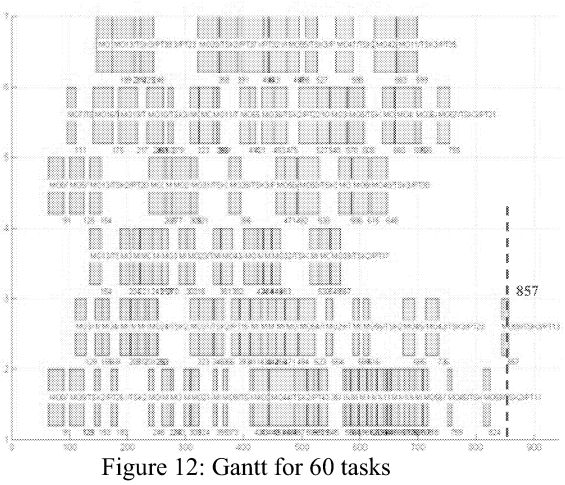


Figure 12: Gantt for 60 tasks

CONCLUSION AND PERSPECTIVES

In this paper, we have presented the promising application of the concept of potential field to the dynamic routing and allocation of products in FMS. Our models have been validated using Netlogo simulation tool. A comparison with an existing benchmark has been made.

Validating this model in a real implementation taking part for the flexible assembly cell of the AIP-PRIMECA Valenciennes will be the next challenge. Potential field will be managed by information systems. The objective will be to compare our results with the real implementation made by (Raileanu 2009) in the same assembly cell.

In our model, decisions of product must be made at decision nodes and potential fields are emitted in every point of the cell. To support this, in our on-going real implementation, a decision node will be composed of a node controller and a data communication system. The node controller will be a 750-841 (Wago 2007). It will oversee the transfer gate, supporting the routing functionalities. Product will communicate to the controller with IrDA system (Clarinet system 2007). In this node controller, the update of potential field values will be done (as a consequence, in a discrete geographical space instead of a continuous one).

REFERENCES

Barraquand J. and Latombe J. 1989, "Robot Motion Planning: A Distibuted Representation Approach". Research Report, STAN-CS-89-1257, Department of Computer Science, Stanford University, 1989.

Berger T., Sallez Y.,Trentesaux D., (2009) "Open control of FMS and its application to potential field", *CIRP09, 42nd Conference on Manufacturing Systems*, Wed. 3 - Fri. 5, June 2009, Grenoble, France.

Clarinet System, Network Connectivity for Mobile Devices, http://www.clarinetsys.com, 2007.

Dorigo, M., and Stützle, T. 2004, Ant Colony optimization. The MIT Press, 2004.

Ferber J. 1995, Les systèmes multi- agents – vers une intelligence collective, InterEditions, Paris (ISBN 2-7296- 0572- X), 1995.

Khatib, O. 1985, Real-time obstacle avoidance for manipulators and mobile robots. *In IEEE international conference on robotics and automation*, pages 500-505, 1985.

Maione, G., and Naso, D. 2003, "A soft computing approach for task contracting in multi-agent manufacturing control". *Computers in Industry*, 52, 199–219, 2003.

Mamei M., Menezes R., Tolksdorf R., Zambonelli F.2006, "Case studies for self-organization in computer science". *Journal of Systems Architecture* 52, 443-460, 2006.

Montech Technology, 2008, Conveyor systems, http://www.montech.com, 2008.

Moujahed S., Approche multi-agents auto-organisée pour la résolution des contraintes spatiales dans les problèmes de positionnement mono et multi-niveaux. Thèse de l'Université de Franche-Comté et de l'Université de Technologie de Belfort-Montbéliard, 2007.

Okino, N. 1993, "Bionic Manufacturing System in Flexible Manufacturing System: past – present – future". J. Peklenik (ed), CIRP, Paris, 73-95, 1993.

Raileanu S., Sallez Y., Beger T., Borangiu T., Trentesaux D., "Holonic implementation of the open-control paradigm". *International conference on Industrial Engineering and Systems Management*, Montreal, 2009.

Sallez Y., Berger T., Trentesaux D., (2009) "A stigmergic approach for dynamic routing of active products in FMS", *Computer In Industry*, 60(3), pp 204-216.

Smith R.G. 1980, The contract net protocol: High level communication and control in a distributed problem solver, *IEEE Transactions on Computer*, vol. C29, n°12, pp. 1104-1113, 1980.

Trentesaux, D., Dindeleux, R. and Tahon, C. 1998, "A MultiCriteria Decision Support System for Dynamic task Allocation in a Distributed Production Activity Control Structure". *Int. J. of Computer Integrated Manufacturing*, 11 (1), 3-17, 1998.

Trentesaux D., Pesin P. and Tahon C. 2000, "Distributed artificial intelligence for FMS scheduling, control and design support". *Journal of Intelligent Manufacturing* 11, 573–589,2000.

Trentesaux, D. 2007, Les systèmes de pilotage hétérarchiques : innovations réelles ou modèles stériles ?. *Journal Européen des Systèmes Automatisés*, 41 (9-10), 1165-1202, 2007.

Ueda K., Vaario J.1998, "An emergent modelling method for dynamic scheduling". *Journal of Intelligent Manufacturing*, pp. 129-140, 1998.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, L. 1998, "Reference architecture for holonic manufacturing systems: PROSA". Computers in Industry, 37 (3), 255–274, 1998.

Wago system, innovative connections, http://www.wago.com, 2007.

Wilensky U. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, 1999.

Zbib N., Raileanu S., Sallez Y., Berger T.,Trentesaux D. 2008, From "Passive Products to Intelligent Products: The Augmentation Module Concept". *5th int. CIRP-sponsored DET conference*, Nantes, France, 22-24th October 2008.

MODELLING AND SIMULATION OF THE GEOMETRY OF CROSS-SECTIONAL VARIATION OF SINGLE YARNS

Ionut Neagu
Pascal Bruniaux
ENSAIT " Ecole Nationale Supérieure
des Arts et Industries Textiles "
2, allée Louise et Victor Champier,
Roubaix, 59056, France.
ionut.neagu@ensait.fr

Dorin Avram
Adrian Rozenberg (Neagu)
Technical university "Gh. Asachi"
Str. D-trie Mangeron nr. 53, Corp
TEX1, Iasi-700050, Romania
dorin.avram@tuiasi.ro
adyneagu@yahoo.com

KEYWORDS

Fiber-to-yarn, simulation interfaces, model design, virtual reality, Computer Aided Design.

INTRODUCTION

The geometric structure of yarns characterizes the yarn from the point of view of fibre position as against the core and axis of the yarn. It was called geometric structure because the analysis of the yarn structure is based on geometric elements of yarns and fibers, such as: yarn diameter, radius position of fiber in yarn, pitch angle of yarns, fiber length, length of a fiber element, surface and volume of fibers and surface and volume of yarns. It is known that fibers have a random position within the yarn, characterized by a variable number of fibers in cross-section and different radial positions along the yarn. As a result, the yarn structure is characterized by analysizing the position of the fibers within the yarn, the fiber stress due to fiber position and the yarn behavior at stress. This can be done based on a theoretical structure which, due to the introduced assumptions, can create the conditions for the mathematical transposition of the phenomena, with a view to analyzing the influence coefficients. Textile fibers have different physical properties of length, count, crimpiness, etc. - so different processes have been designed by technologists to optimize the production efficiency of yarn for a particular fiber type and end use. Consequently, different types of yarns may vary in appearance and mechanical properties. A useful yarn model is certainly a valuable tool for yarn technologists involved in the product development process. Much effort can be saved by avoiding the trial and error approach. According to Hearle, the research done until now in the field of fiber simulation has two main drawbacks: the lack of a generalized model, on the one hand, and the absence of a model that combines one or more hierarchical models, on the other hand. Many research papers focus on the mechanical modelling of the geometric structure of yarns, but there is no model which takes into consideration the cause-effect of fiber integration in the yarn structure. One of the most important yarn features is the fiber uniformity which is influenced by a series a coefficients such as the technological process and the number of fibers in the cross-section, so that fiber attenuation determines yarn resistance. The variations of yarn parameters such as diameter, torsion, resistance etc. cannot be avoided, although this is an undesired situation in most cases. Dorin Avram states that the variation of yarn characteristics may cause problems during and after the manufacturing process.

The diameter - count relationship has been studied by Adanur and others, who state that it depends on the specific volume of the yarn. Irregularity is a major yarn feature which determines its appearance and mechanical characteristics. Choi comes up with an interesting assumption relating to the connection between fiber length and yarn diameter variation, which influences both the location of yarn axis and the accuracy of migration parameters. Since textiles have a hierarchical structure, we cannot validate and simulate such a structure without taking into consideration the cross-sectional variation of diameters, which is determined by the irregularity of yarn fibers. The latest studies assert that the variation of diameters during the spinning process is the most important feature. According to a study carried out by “Australian Innovation Pty Ltd”, a diameter variation limited to 15% leads to the improvement of subsequent manufacturing processes. Another paper, that of Moon, Jasper and Woo focuses on the simulation of yarn characteristics. Its main purpose is to identify yarn irregularity by using a technology that combines optical and capacitive sensing devices. Kılıç and others have studied the relations between yarn irregularity and transversal forces. Pan focused on the relations between fiber forces and yarn forces. Kremenakova perfected a research methodology for the simulation of yarn structure and properties. Another study that analyses the simulation models using a structure for image recognition was carried out by Kılıç and Okur. Robitaille & Sherburn have used a morphing model, but disregarded the role of fiber into the yarn geometry. Other studies by Bruniaux and others reveal the interdependence between yarn and fabric, because the yarn structure has to be used in a fabric structure this studies show us what should be the first starting point of a yarn geometry.

THE MODELLING ALGORITHM

Average count of fiber.

$$\overline{Nm}_f = \sum a_i \cdot Nm_{fi} \quad (1)$$

Where:

a_i - The mass percentage of i component;

Nm_{fi} - Average count of fibres of i component.

The limit variation coefficient is determined under the assumption that the fibres in the yarn section vary according to the Poisson law.

The limit variation coefficient for the yarns in heterogeneous blends is determined by the relation:

$$CV_{\lim} = \sqrt{a_1 \cdot CV_{\lim 1}{}^2 + a_2 \cdot CV_{\lim 2}{}^2} \quad (2)$$

Where:

$CV_{\lim 1}$ - The limit variation coefficient of the yarn for component 1;

$CV_{\lim 2}$ - The limit variation coefficient of the yarn for component 1;

The limit variation coefficient for every component is determined by the relation:

$$CV_{\lim} = \frac{100}{\sqrt{n_s}} \quad (3)$$

Where:

n_s - The number of fibres in the cross-section of the yarn of each component.

$$n_{s1} = \frac{Nm_{f1}}{Nm} \quad (4)$$

$$n_{s2} = \frac{Nm_{f2}}{Nm} \quad (5)$$

Average fibre density (g/cm3).

$$\overline{\rho_f} = \frac{1}{\frac{a1}{\rho_{f1}} + \frac{a2}{\rho_{f_i}2}} \quad (6)$$

Where:

a_i - The mass percentage of i component;

ρ_{fi} - Average fibre density of i component.

Average yarn density (g/cm^3).

$$\rho_F = \nu \cdot \overline{\rho_f} \quad (7)$$

$$\nu = f(T)$$

Where:

ν - is adopted between 0.6 and 0.7;

$\overline{\rho_f}$ - Average fibre density.

The yarn diameter varies statistically according to a normal distribution and can be illustrated as follows:

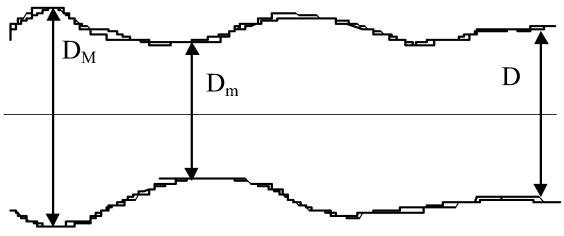


Figure 1: Statistic diameter variation

Average diameter of the yarn (μm).

$$D = \sqrt{\frac{4}{\pi \cdot Nm \cdot \rho_F}} \quad (8)$$

Interdependence between yarn diameter and the number of fibres in the yarn section.

Knowing from relation 7 the average density of fibres:

$$\overline{\rho_f} = \frac{4}{\pi \cdot Nmf \cdot d^2} \quad (9)$$

Where:

$\overline{\rho_f}$ - Average fibre density;

Nmf - Average fibre count;

d - Fibre diameter.

Let us replace the average fibre density with equation 9, thus obtaining the equation:

$$\rho_F = \nu \cdot \frac{4}{4Nmf \cdot d^2} \quad (10)$$

Knowing the yarn diameter equation and substituting yarn density with equation 10, we have:

$$D = \sqrt{\frac{4}{\pi \cdot Nm \cdot \nu \cdot \frac{4}{\pi \cdot Nmf \cdot d^2}}} \quad (11)$$

Thus equation 11 becomes:

$$D = \sqrt{\frac{Nmf}{Nm \cdot \nu} d^2} \quad (12)$$

Knowing that the equation of the number of fibres in the yarn cross-section is:

$$ns = \frac{Nmf}{Nm} \quad (13)$$

From equation 12, replacing the number of fibres in the yarn cross-section, we have:

$$D = d\sqrt{\frac{ns}{\nu}} \quad (14)$$

The mean square deviation is determined by the relation:

$$\sigma = \frac{CV_{ef} \cdot D}{100} \quad (15)$$

CV_{ef} - is the effective variation coefficient of yarn diameter and is determined by the relation:

$$CV_{ef} = CV_{\lim} \cdot I \quad (16)$$

Where:

I is the irregularity coefficient and we adopt between I = 1.2 – 1.3.

The computing of diameter limits.

$$D_{\min} = D - 1{,}96 * \sigma$$
$$D_{\max} = D + 1{,}96 * \sigma \quad (17)$$

Where:

D - Average yarn diameter;

σ - Standard deviation of yarn diameter.

The diagram of yarn diameter variation is obtained as follows:

We determine the class amplitude using the relation:

$$a = \frac{D_{\max} - D_{\min}}{n_c} \quad (18)$$

n_c -is the number of classes (20)

The diameter varies according to a normal function whose distribution density is:

$$f(D_i)=\frac{1}{\sigma\cdot\sqrt{2\cdot\pi}}\cdot e^{-\frac{1}{2}\left(\frac{Di-\overline{D}}{\sigma}\right)^2} \quad (19)$$

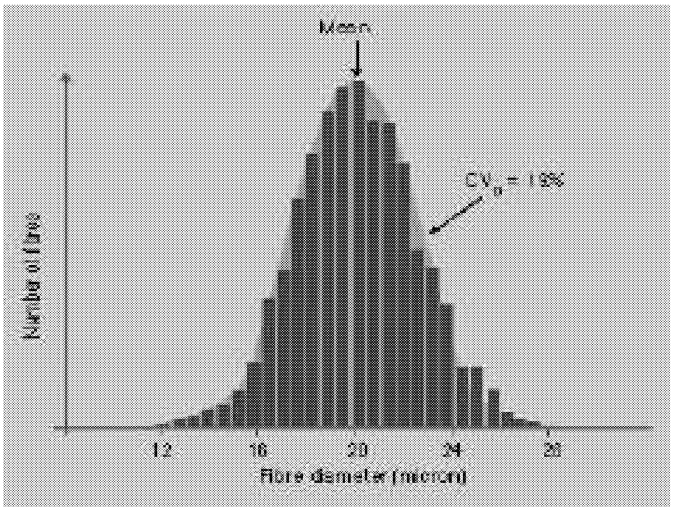


Figure 2: Diameter distribution bar chart

The yarn diameter was considered to follow a normal distribution in the case of a large number of millimeter to millimeter measurements (over 100 values). We estimated that the distribution of the diameter (continuous distribution) was similar to the variation of the number of fibres in the cross-section of the yarn (discontinuous distribution). The diameter limits have been established for a statistic safety of 95%, under the assumption of normal distribution.

BEZIER POLYNOMIALS

Bezier polynomials are used for the approximation of the plane curve implemented to unite the generated diameters of the yarn in order to produce the exterior appearance thereof. One of the major problems related to object design is shape design. When it comes to generating simple forms, we can use analytic entities such as dots, line segments, circles, circular sectors, conics, etc. As far as complex forms are concerned, this type of representation has the disadvantage that any modification of the object, e.g. by rotation, scaling or distortion, is almost impossible to achieve. For this reason, it is easier to place some reference dots and generate curves among them so as to approximate the form of the object. Bezier curves represent this type of synthetic entities.
A parametric curve is defined by a discrete set of dots known as control dots together with a set of basic or combination functions. The basic functions used to place and arrange the control dots in a curve segment may be represented by any set made up of a number of basic functions with different properties used in computer graphics to control the shape of a curve through an interactive interface. Bezier curves are successfully used due to their simplicity and because they can be easily represented when Bezier surfaces are used.

HOW THE PROGRAM WORKS

What is computer simulation?
Virtual objects that simulate the behaviour of simulated objects in a real way, whose aspect and behaviour correspond to that of a simulated object.
SIMUDIM is a computer program that simulates the diameter of woollen spun carpet yarns. The program was created in the PERL programming language. The main panel and the 3D viewer windows are presented in the figure 3, with the input panel in the upper left side and the output panel below.

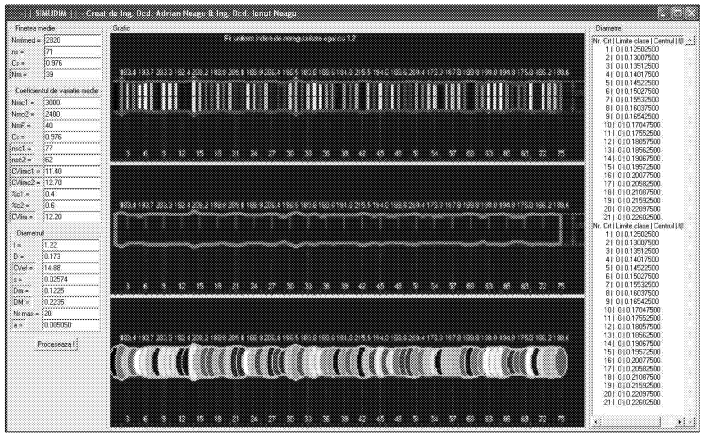


Figure 3: The main panel and the 3D viewer window

Working algorithm of the program
Average characteristics, i.e. the average yarn diameter obtained through experimental sampling, are input into the program. The variable data that is inserted is presented in table 1.
The software automatically presents a table of the generated diameters used in the simulation based on the medium diameter, as seen in figure 4.

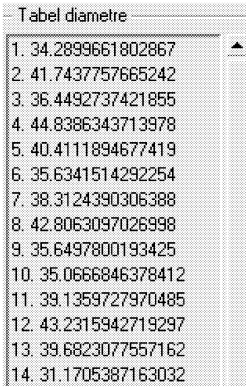


Figure 4: Table containing the values of the generated diameters

Depending on the yarn irregularity, the program displays two variables: a) when the irregularity index is $I \leq 1.22$, the message in figure 5 is displayed, and b) when the irregularity index is $I \geq 1.22$, the message in figure 6 is displayed.

Even yarn I equal 1.22

Figure 5: The uniformity message

Unevenness yarn I equal 1.27

Figure 6: The irregularity message

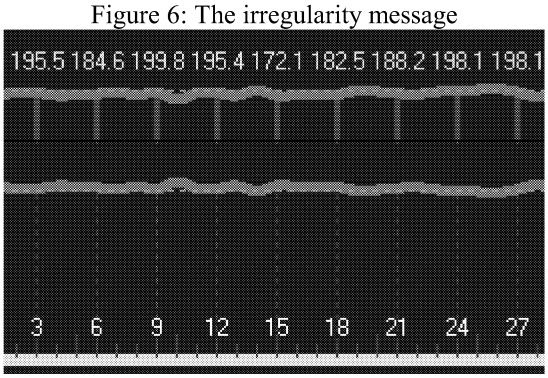


Figure 7: The display of generated diameters (μm).

Table 1 presents the input data of yarns and fibres characteristics, while table 2 shows the output data.

Table 1: Input data

Input data The	UM	Desig natio n
A1 is the mass percentage of the component 1	%	*a1*
A2 is the mass percentage of the component 2	%	*a2*
Average fibre count of the first component;	m/g	*Nmf1*
Average fibre count of the second component	m/g	*Nmf1*
Average yarn count;	m/g	*Nm*
Average fibre density of the i component.	g/cm^3	ρ_{fi}
Irregularity index	-	*I*
Packing rate	-	*v*
Fibre diameter	μm	*d*
Number of classes	-	*nc*

Table 2: Output data

Output data	UM	Not.
Average count of yarn fibres;	m/g	$\overline{Nm_f}$
Average fibre density;	g/cm^3	$\overline{\rho_f}$
Yarn density;	g/cm^3	ρ_F
Average yarn diameter;	μm	*D*
Mean square deviation of yarn diameter.	-	σ
Maximum diameter of the yarn.	μm	D_{max}
Minimum diameter of the yarn.	μm	D_{min}
Limit variation coefficient of i component.	%	CV_{limi}
Limit variation coefficient of the yarn diameter	%	CV_{lim}
Effective variation coefficient of the yarn diameter.	%	CV_{lim}
Class amplitude.	-	*a*

In order to exemplify how the program runs, we have chosen three types of yarns, all having the same count of 40 Nm and average diameter, but different irregularity coefficients See input data displayed in table 3, and therefore the output is presented in table 4.

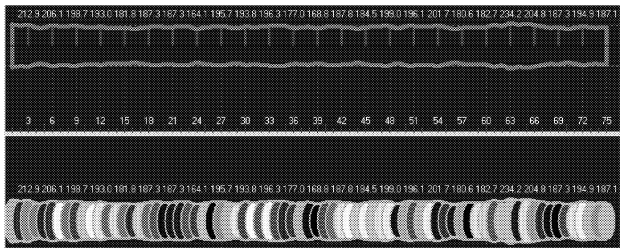


Figure 8: Variation of yarn diameter of 40 Nm count and I=1.2.

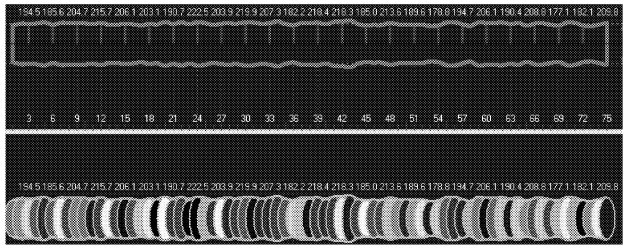


Figure 9: Variation of yarn diameter of 40 Nm count and I=1.25.

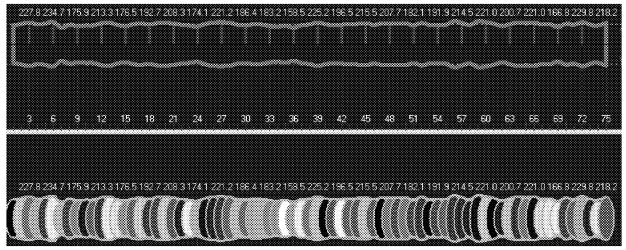


Figure 10: Variation of yarn diameter of 40 Nm count and I=1.3.

Table 3 Characteristics of the analyzed yarns

	UM	V1	V2	V3
A1 is the mass percentage component 1;	%	40	40	40
A2 is the mass percentage component 2;	%	60	60	60
Average fibre count of the component 1;	m/g	3000	3000	3000
Average fibre count of the component 2;	m/g	2400	2400	2400
Average yarn count;	m/g	39	39	39
Average fibre density of the i component;	g/cm^3	1.38	1.38	1.38
Irregularity coefficient;	-	1.52	1.52	1.52

Packing rate;	-	1.2	1.25	1.3
Fibre diameter;	μm	0.6	0.6	0.6
Number of classes.	-	20	20	20

Table 4 Output data for the analyzed yarns

	UM	V1	V2	V3
Average count of yarn fibres;	m/g	2820	2820	2820
Average fibre density;	g/cm^3	1,46	1,46	1,46
Yarn density;	g/cm^3	0,43	0,43	0,43
Average yarn diameter;	μm	0,27	0,27	0,27
Standard deviation of yarn diameter;	-	0.033	0.045	0.051
Minimum diameter of the yarn;	μm	0.203	0.19	0.016
Maximum diameter of the yarn;	μm	0.336	0.349	0.379
Limit variation coefficient of i component;	%	0.006	0.007	0.010
Limit variation coefficient of the yarn diameter;	%	2820	2820	2820
Effective variation coefficient of the yarn diameter;	%	1,46	1,46	1,46
Class amplitude.	-	0,43	0,43	0,43

CONCLUSIONS

Despite the numerous studies on the modelling and simulation of textiles, none focuses on the dynamic behaviour of the yarn, i.e. none takes into consideration fibre characteristics in terms of modelling the yarn structure. Since the yarn structure is based upon the average characteristics of yarns and fibres, such as yarn diameter, fibre diameter, average yarn count, average fibre count, fibre density, participation percentage, the simulation is meant to be a statistic approach due to the normal distribution of yarn diameters. We chose to implement the statistic distribution because it is closer to reality from the modelling point of view, knowing that the average of total population can be determined using the measurements obtained during a single test. It can be noticed that the standard deviation decreases as the number n of test experiments increases. Another distribution that was approached in this simulation was that of Weibulldist, but it is irrelevant since modelling is very difficult due to the large deviations of results.

The use of interpolation curves represented by Bezier polynomials is a new approach, especially in the field of cross-sectional variation of woollen spun carpet yarns. Our main goal is to implement the B-spline curves since Bezier curves present some disadvantages that can be eliminated by using B-spline curves (cubic curves are sometimes used). This type of curves is flexible enough to be used in most applications at lower costs and represent, in fact, real spatial curves, unlike the square ones which are contained in a flat surface. A curve made of square segments is but a set of bi-dimensional segments. The textile industry has new advantages that result from the use of yarn modelling, i.e. 3D development of textile products, innovative product design for enhanced competitivity, improved product quality, shorter delays in product delivery and an optimum cost control. This paper aims to complete the research done during the textile modelling and simulation applications. By using 3D tools, companies can reduce the time required for product design. The immediate result is the reduction of working hours and of costs. Thanks to the development of more affordable computer technologies, virtual tools are implemented, which contribute to the improvement of a realistic and technical level of simulation quality.

In future, research work should be homogenised for use in industry.

REFERENCES

Avram Dorin, 2004 "Yarns' Structure", Iasi, ISBN 973-7994-46-9,

Bruniaux P. & J.M.Castelain , 2003, "Traction model - A methodological approach", ISC'2003 Conference Workshops on Modelling and Simulation in the Textile Industry, pp. 523-530, Valencia, Spain, 9-12 June.

Bruniaux P., X.Legrand, C.Ngo Ngoc & J.M.Castelain 2002 "Modélisation mésoscopique du tissu par le fil ", XIIth Romanian Conference of Textiles and Leadership, Iasi, Roumanie.

Bruniaux P., X.Legrand & C.Vasseur, 2006a, "Yarn and fabric model inter-connected", Journal of Advanced Materials, Vol. 37, N°4, pp 60-69.

Bruniaux P. & C.Ngo Ngoc, 2006b, "Nonlinear model of a fabric warp and weft", Advances in Complex Systems, Vol. 9, N°1-2,

Dana Kremenakova, "Methods for investigation of yarnstructure and properties Computer Aided Textile Design"

François Robitaille & Martin Sherburn,"Modelling the geometry of textile structures for property prediction"

K Choi, "Fiber Segment Length Distribution on the Yarn Surface in Relation to Yarn Abrasion Resistance"

Musa Kılıç & Ayşe Okur, "Relationships Between Yarn Diameter / Diameter Variation and Strength"

Sabit Adanur,Yehia El Mogahzy, Faissal Hady, "Yarn and fabric design and analysis system in 3D virtual reality"

Kirill E Perepelkin, "New Impulse Acoustic Method for Testing and Estimating the Transversal Unevenness and Defectiveness of Filaments in Chemical Yarns"

Ning Pan, "Relationship between fiber and yarn strength" pp 99-120.

Modeling of a Pump-less LiBr Absorption System for Residential Consumption

H. Monsef [1], M. Naghashzadegan[2], M. Farshad Manesh[2], S. Roohi[3]
[1]Master student, Mohaghegh University, Iran
[2]Assistance professor, Guilan University, Iran
[3] Master student, Guilan University, Iran

Keywords: LiBr Absorption system, Bubble Pump, Design and Construction, Residential consumption

Abstract
The objective of this paper is to present a method for design and construction of a low capacity LiBr absorption system for residential consumption. At this system mechanical pump has been replaced with a bubble pump for reducing the cost of the system. Several test rig bubble pumps have been built with a single pipe Pyrex with different diameters and submergence ratios to find the best parameter of the bubble pump. Results of experiments show that the bubble pump with 2.5 mm diameter and submergence ratio of 0.4 is suitable for using in absorption system. Also analytical results have shown that for an absorption refrigeration system with a capacity of 2.5 KW a bubble pump with five horizontal tubes had the best performance.

Introduction
A conventional vapor absorption cycle is running primarily on heat energy, requiring a mechanical energy to circulate the refrigerant–absorbent solution from the low-pressure absorber to the high-pressure generator. Duo to high cost of mechanical pump their use in residential consumption are very rear.
Platen and Munters [1] studied the diffusion absorption refrigerating cycle, where the circulation of the working fluids is carried out by a bubble pump.
A bubble pump is a thermal fluid pump, using thermal energy to pump liquid from lower level to the higher level. At the bottom of the pump tube, the generated small bubbles join together forming bigger vapor bubbles. The rising vapor bubble acts like a piston and lifts a corresponding liquid slug to the top of the bubble pump tube.
Comparing with the conventional absorption cycle, the diffusion absorption refrigeration system had the following advantages:
1. Silent operation.
2. No moving parts: no fans or pumps, the cycle is very reliable.
3. Portability: the unit can utilize any heat source (above $150^{\circ}C$) and therefore can operate anywhere, even in countries without electric utility infrastructure.
Chen et al. [2] developed a new generator configuration where increased the COP of the cycle by 50%. The generator consists of heating elements, a bubble pump and a coaxial heat exchanger. This configuration allowed the reduction of heat losses, increasing the efficiency of the heating process and increasing the COP.
In order to validate the mathematical model, Pfaff et al. [3] developed a mathematical model using the manometer principle to evaluate the bubble pump's performance. A test rig was built in glass to evaluate the bubble pump performance and to visualize the flow patterns. Pfaff et al. built the test rig with several bubble pumps, connected in parallel to compare the performance of the bubble pump with different tube diameters. They have found experimentally that the bubble pump operated at slug flow regime in cyclic intervals. They have shown that the pumping ratio (the ratio between the strong solution flow rate and the refrigerant flow rate) is independent of the heat input. They have also found that a bubble pump with a tube diameter of 10 mm and heat supply of 40W is suitable for cooling capacity of about 100 W.
Delano [4, 5] dealt with design analysis of the Einstein refrigeration cycle. This cycle incorporates a bubble pump to circulate the working fluids and utilizes three-component working fluid, water–ammonia–butane, water as absorbent, butane as the refrigerant and ammonia as the auxiliary gas. He developed an analytical model to evaluate the bubble pump performance. Using the analytical model, the influence of heat input, tube diameter and submergence ratio on bubble pump performance has been investigated.
He found that increasing the heat input to the bubble pump for a fixed submergence ratio, increases the flow rate of the liquid through the bubble pump to a maximum value where further increase in the heat input, decreases the liquid flow rate. He showed that increasing the tube diameter, decreases the friction factor and therefore increases the flow rate through the bubble pump.
Koyfman et al. [6 ,7] presented a continuous experim-ental system, suing a practical working fluid. The flow was induced by boiling of a binary mixture (as in a real diffusion absorption refrigerating unit). This experim-ental system allows changes in the system pressure as well as changes in the systems geometry.
Am absorption system can be classified according to the way of heating its generator and it is possible the cycle be one step or more. One step systems work with a direct flame and for high heat value of fuel, their COP is about 0.5 [8].Temperature range of generator of Lithium Bromide-Water cycles is from 75 °C to 120 °C [9]. More steps systems result higher COP, however only when heat sources with high temperature are achievable.

Absorption systems
A one step Lithium Bromide-water system with mechanical pump has shown in figure 1. The pressure-temperature diagram of an absorption system with mechanical pump has been shown in figure 2.

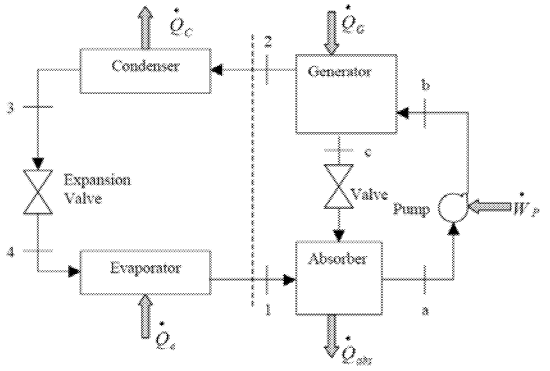


Figure 1: A schematic of LiBr-water absorption system with a mechanical pump

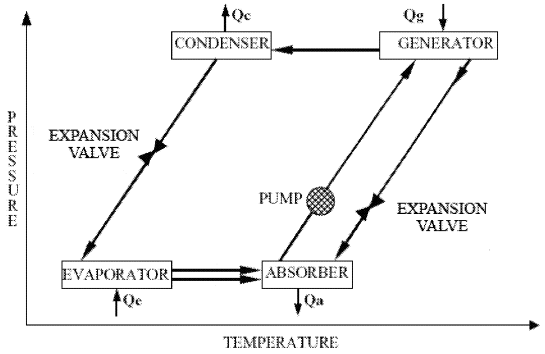


Figure 2: Pressure- temperature diagram of LiBr-water absorption system with a mechanical pump

A mechanical pump replaced with bubble pump to reduce the cost and noise of the system. Using bubble pump also eliminates the need for electricity power. A schematic view of bubble pump has been shown in figure 3.

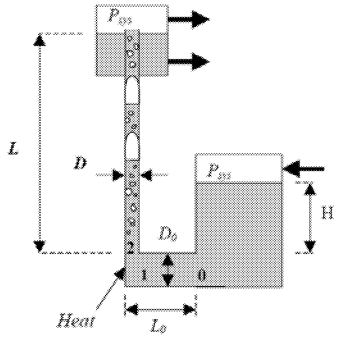


Figure 3: A schematic of bubble pump

Several factors such as diameter of vertical tubs, submergence ratio and input heat affect the height and the flow rate of pumped fluid. For analyzing these parameters, several experimentally single pipe Pyrex bubble pump has been built and tested with various diameters, submergence ratio (H/L) and input heat.
Experimental results show that the vapor flow rate increases to a maximum value as the input heat increases, where it reaches a steady value at about 800 (w), where, the flow regime has been changed to churn and annular from slug regime [10]. Figure 4 shows the outlet vapor flow rate from bubble pump versus internal heat power for different pipe diameters.

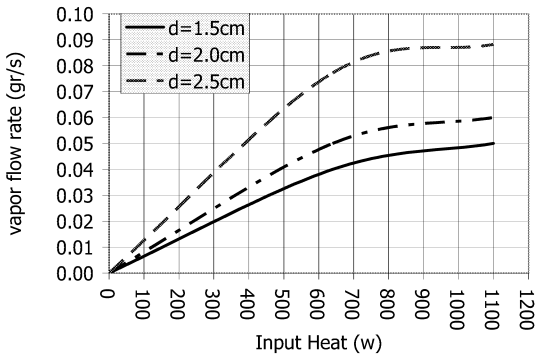


Figure 4: Variation of vapor flow rate versus internal heat

Figure 5 shows the variation of liquid flow rate versus of the input heat. It is seen that as the input heat increases, the liquid flow rate increases too. The results show that the liquid flow rate in the thinner tubes is more than that vapor flow rate and that is because in thinner tubes, the bubbles can fill tubes easier and push the fluid upward, acting like a piston.

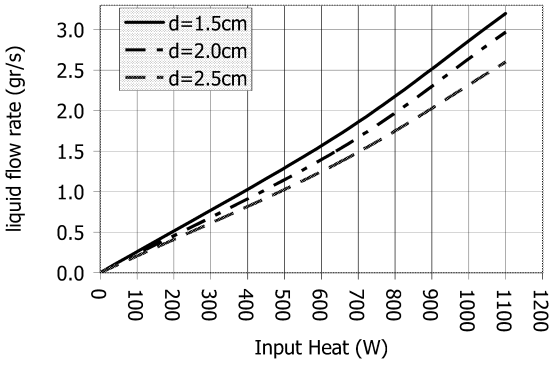


Figure 5: Variation of liquid flow rate versus internal heat

Figure 6 compare the effect of submergence ratio (H/L) on the liquid flow rate. The Figure shows that the larger H/L has the larger liquid flow rate. At H/L less than 0.2 the bubbles reach to the surface of fluid rapidly and that is because enough fluid doesn't exist between the bubbles.

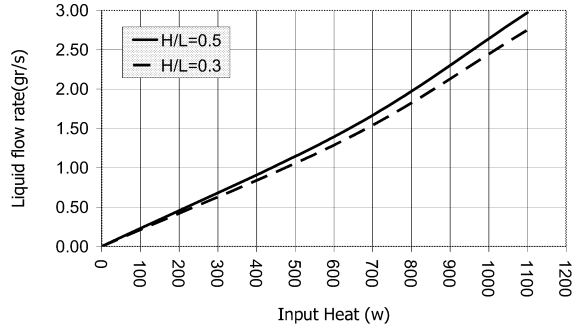


Figure 6: Effect of submergence ratio (H/L) on liquid flow

According to the experiments, the vapor flow rate is increased in larger tubes than in the thinner tube. is the

reason is that in the LiBr refrigeration system the refrigerant is in vapor status and using the larger tubes in bubble pump therefore is more desirable.
The performance of absorption systems with bubble pump is exactly similar to the performance of systems with mechanical pump, where bubble pump increases the pressure of fluid and circulate the fluid in the cycle. Figure 7 shows a schematic diagram of a LiBr absorption system using a bubble pump. Pressure has been increased due to the lift of fluid in bubble pump and heat of generator.

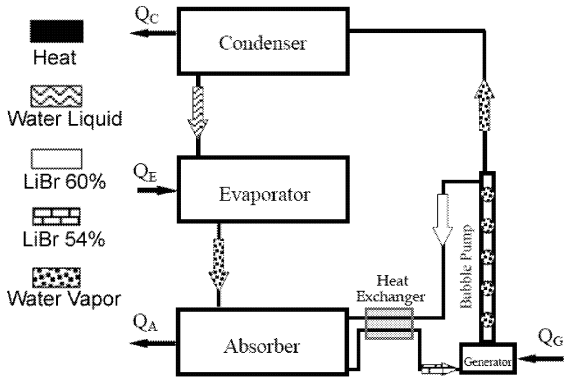


Figure 8: A schematic diagram of a LiBr absorption system with bubble pump

Experimental Setup
The amount of refrigerant (water) should be calculated to build an absorption refrigerant system with low capacity. Equation (1) calculates the amount of water circulating in the system.

$$Q = mh_{fg} \tag{1}$$

For a system with capacity of $2.5\,kw$, the required flow rate is about $1.075\,gr/s$. Experimental results show that a bubble pump with five tubes of $2.5\,cm$ diameter and $H/L = 0.4$ can produce this flow rate.
A five tubes steel bubble pump has been made where generator and bubble pump are united in this system.
Vertical tubes with low inner friction are used to increase the efficiency of pump. The tubes are isolated to reduce the heat loss. The heat transfer between bubbles and surrounding fluid will reduce the size of bubbles. Where small bubbles can't lift above fluid and finally will be exploded.
Increasing input heat in generator increases the pressure of condenser which has two disadvantages:
1- The pumping of LiBr solution will be reduced, because the pressure at upper tank has been increased.
2- Increasing pressure of condenser will stop refrigeration cycle due to disturb pressure balance in cycle. Thus the input heat in generator must have an optimum amount. Figure 9 shows the absorption refrigeration system test rig.

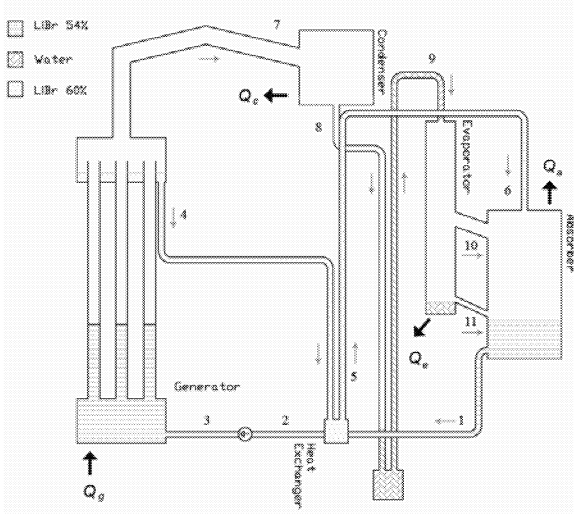


Figure 9: Schematic of built absorption refrigeration system

Condenser
A shell and tube condenser is used where vapor flows in shell and cooler fluid in copper tubes. The condenser performance is given by energy equilibrium as equation (2). Points are shown in figure 9.

$$\dot{Q}_{con} = \dot{m}_7 (h_7 - h_8) \tag{2}$$

The total heat transfer rate $\dot{Q}$ between the fluids can be determined from the following simple expression:

$$\dot{Q} = AU\ \Delta T_m \tag{3}$$

Where

$$\Delta T_m = F\,\Delta T_{\ln} = F\left(\frac{\Delta T_0 - \Delta T_L}{\ln(\Delta T_0 / \Delta T_L)}\right) \tag{4}$$

The overall heat transfer coefficient (U) based on the outside surface of the tube is defined as [11]:

$$U = \frac{1}{(\frac{D_o}{D_i})(\frac{1}{h_i}) + (\frac{D_o}{D_i})F_i + (\frac{1}{2k})D_o \ln(\frac{D_o}{D_i}) + F_o + \frac{1}{h_o}} \tag{5}$$

The value of the fouling factors (F_i, F_o) at the inside and outside surfaces of the tube can be taken as $0.09\,m^2K/kW$ [12]. For a copper heat exchanger, k can be calculated by curve fitting method [11]
Equation (5) has been used in a computer program for designing the shell and tube condenser.
For this research, the cooling water inlet and outlet temperatures must be assumed as input.
Assuming a turbulent flow inside the heat transfer coefficients (h_o, h_i) for the inside and outside flow is given by [13]:

$$\overline{Nu}_D = \frac{\frac{f}{8} Re_D Pr}{K_1 + K_2 \left(\frac{f}{8}\right)^{1/2} \left(Pr^{2/3} - 1\right)} \quad (6)$$

Where

$f = (1.82 \log_{10} Re_D - 1.64)^{-2}$ ، $K_1 = 1 + 3.4 f$,

$K_2 = 11.7 + (1.8 / Pr^{1/3})$

Eq. (6) applies for $10^4 < Re_D < 5 \times 10^6$ and $0.5 < Pr < 2000$ respectively.

The water properties are evaluated at a mean temperature. Nusselt's analysis of heat transfer for condensation on the outside surface of a horizontal tube gives the average heat transfer coefficient as [14]:

$$h_m = 0.725 \left[\frac{g \rho_l (\rho_l - \rho_v) h_{fg} k_l^3}{\mu_l (T_v - T_w) D_o} \right]^{0.25} \quad (7)$$

The logarithmic mean temperature difference has been used to evaluate the necessary heat transfer area of the tube.

Heat Exchanger

One of most important point to increase efficiency of system is selecting a suitable heat exchanger. The heat exchanger warm the weak solution, coming from absorber, reducing temperature of strong solution coming from bubble pump before entering to absorber. Lower temperature of strong solution helps to absorb vapor easier.

Eq. (5) is used to calculate the overall heat transfer coefficient. The value of the fouling factors (F_i, F_o) for the inside and outside surfaces of the tube are taken as $0.09\, m^2 K/kW$ [12] and k for copper will be evaluated at a mean temperature. The heat transfer coefficients h_o, h_i for the inside and outside flow are determined as detailed below.

The LiBr solution properties at a mean temperature are calculated and Re is determined from

$$Re = \frac{4\dot{m}}{\pi D \mu} \quad (8)$$

For laminar flow, the Nusselt number is given [12]:

$$Nu = h_i D_i / k = 3.66 \quad (9)$$

The hydraulic diameter D_H for the annulus is the difference between the inside diameter of the external tube D_2 and the outside diameter of the internal tube D_1 .

The Reynolds number based on the hydraulic diameter and the bulk temperature properties is:

$$Re = \frac{\rho V \overline{D}_H}{\mu} = \frac{\dot{m} D_H}{4\mu} \quad (10)$$

For laminar flow, as before $Nu = h_o D / k = 3.66$ [15], and therefore, h_o may be evaluated.

By substituting the above values in Eq. (5), the resulting overall heat transfer coefficient (U) based on the outside surface of the tube is subsequently evaluated.

The logarithmic mean temperature is then calculated, and finally, where using Eq. (3), the area of the heat exchanger is determined.

Evaporator

Evaporator has been designed similar to condenser where the external heat transfer coefficient should be calculated by Jacob-Havkinz formula .[16]

$$h_o = h_1 \times (\frac{P}{P_1})^{0.4} \quad (11)$$

Here P is the evaporator pressure, P_1 is atmospheric pressure and h_1 is convection coefficient in atmospheric pressure.

Results have shown a $0.4\ m^2$ of copper surface for system‘s evaporator is necessary. One of the most important parameters in evaporator is the contact area between condensed water (from condenser) and cooper tubes. The velocity of water particles within tubes is important there it has been increased to increase the velocity of water. Figure 9 shows a schematic of built evaporator.

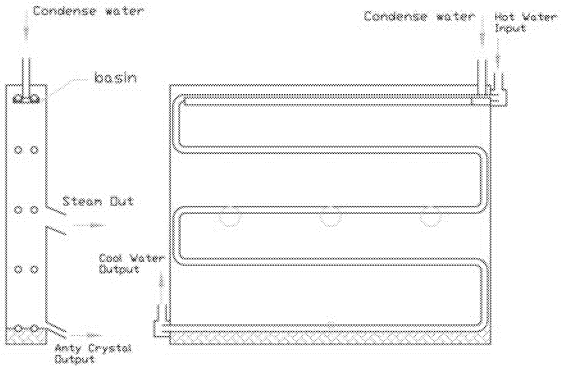


Figure 9: Designed evaporator sketch

Absorber

Absorber is one of importance part of an absorption refrigeration system. In an absorber strong solutions, coming from bubble pump absorb water vapor coming from evaporator. Figure 10 shows three governor regimes in strong LiBr solution on copper tubes.

(1) Falling film in contact with the tube,

(2) Drop formation at the bottom of the tube,

(3) Drop fall between the tubes.

Icksoo Kyung et. al [17] developed a model for absorption of water vapor into aqueous LiBr flowing over a horizontal smooth tube. The results have been used to build an absorber for this research.

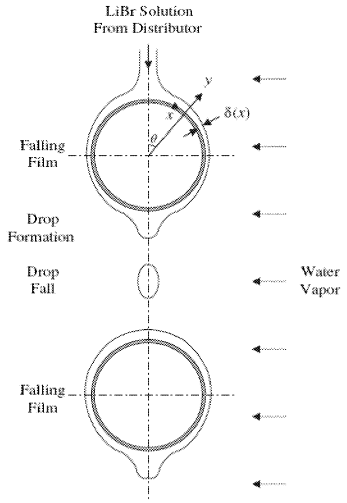


Figure 10: Three applied regime of strong solution of LiBr on horizontal tube

Figure 11 shows a schematic of the experimental system built at Guilan University with capacity of 2.5 *kw* .

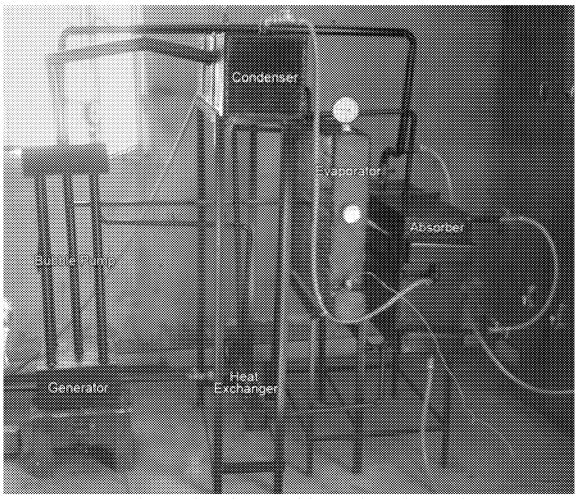


Figure 11: Schematic of our refrigeration absorption System

The machine built with steel and connected to a vacuum pump to decrease system pressure before cycle testing. Manual and digital gage install on system to measure temperature and pressure of all points of system. Heat source in generator is natural gas.

The cycle works at two levels of pressures (0.93 *kPa* and 9.66 *kPa*). If the pressures are matched, the refrigerant system will work properly. In this system, a U-shape tube used between condenser and evaporator. Since the pressure difference is 8.73 *kPa* , a U-shape tube with 90 *cm* heights can produce this pressure difference. Between bubble pump and absorber also another U-shape tube was used which has the same behavior a U-shape tube with 55 *cm* heights can produce this pressure difference. The density of strong Lithium Bromide solution in this point is given by Equation 12 which is about 1600 kg/m^3 ,

Density of LiBr solution [18]:

$$
\begin{aligned}
&Range \;\; 20\% < X < 60\% \; LiBr \\
&T = solution\ temperature\ (^oC), range\, 0 < T_{sol} < 200 \\
&\rho_x = LiBr\ solution density\ (kg/m^3) \\
&X_0 = X/100 \\
&\rho_x = 1145.36 + 470.48\,X_0 + 1374.79\,X_0^2 - \\
&\qquad (0.333393 + 0.571749\,X_0)(273 + T)
\end{aligned}
\tag{12}
$$

A one way valve was used between absorber and generator which at the start of pumping ,prevents returning bubbles to absorber. Since the volume of generator is fixed, as soon as some solution pumped, it will be vacuumed and let the weak solution to be pulled from absorber to generator.

The fluid isn't pumped continuously, because after completing the first cycle, it takes time to heat the solution and produces bubbles again. Because the system's pressure is too low comparing with that of atmospheric pressure, these periods are too short and the fluid is pumped in lower intervals.

Experimental Results

The experimental results are shown in tables 1 and 2. It good agreement with Results of G.A. Florides et al. [15] researches and experiments.

Table 1: Fluid state in different points of cycle

Point	$\dot{m}$ (g/s)	P (kPa)	T ($^\circ C$)	X (%LiBr)
1	12.1	0.93	35	54
2	12.1	0.93	55	54
3	12.1	9.66	55	54
4	11.03	9.66	85	59
5	11.03	9.66	65	59
6	11.03	0.93	65	59
7	1.075	9.66	80	0
8	1.075	9.66	42	0
9	1.075	0.93	6	0
10	1	0.93	6	0
11	0.075	0.93	6	0

Table 2: Heat balance of different part of cycle

Description	Symbol	kW
Capacity (evaporator output power)	$\dot{Q}_e$	2.5
Absorber heat, rejected to the	$\dot{Q}_a$	3.6

environment		
Heat input to the generator	$\dot{Q}_g$	4.9
Condenser heat, rejected to the environment	$\dot{Q}_c$	2.8
Coefficient of performance	COP	0.51

The data has been read from K kind thermocouples and pressure gages that all of them have been calibration before testing. The rate of heat in evaporator, condenser and absorber has been measured with temperature difference between inlet and outlet solution. Also the concentration of LiBr solution has been measured by density meter bulb and the input heat in generator has been measured with internal energy of natural gas and its flow rate.

Low pressure of generator causing weak solution to boil in low temperature (in atmosphere pressure it boil in $145^{o}C$ but now it boil in $80^{o}C$).

High density of strong solution can absorb vapor of air until more water can be vaporize in evaporator.

One of the big problems of LiBr low capacity refrigeration system is water type condenser. These types of condensers needs cooling tower which is for residential consumption unacceptable. The low pressure don't allow use of air kind condensers because, the amount of pressure isn't enough to overcome the friction power in the condenser pipes in air type condenser.

Conclusion

In this research, a test rig was built using a single tube bubble to find the best diameter and submergence ratio for use in absorption systems. According to the primary results, a five steel vertical tube babble pump was built which latter was used in a low capacity ($2.5\,KW$) absorption system. An evaporator, absorber, condenser and a heat exchanger were designed and constructed. The system doesn't used electrical energy, therefore where this energy has high cost the system is affordable.

The generators temperature is ($80^{\circ}C$) which make it possible to use a solar cell. Calculations have shown, the primary cost of this system is higher from ordinary absorption refrigeration systems.

The bubble pump and generator shape play important role in COP of system. Bubble pump of system have to be tested to find liquid and vapor flow rate for each new sketch or other capacity. Generator input heat is almost an important parameter to balance the cycle pressure and pumping solution.

References

[1] B.C. von Platen, C.G. Munters, Refrigerator, US Patent 1,685,764, 1928.

[2] J. Chen, K.J. Kin, K.E. Herold, Performance enhancement of a diffusion–absorption refrigerator, International Journal of Refrigeration 19 (3) 208–218, 1996.

[3] M. Pfaff, R. Saravanan, M.P. Maiya, M. Srinivasa, Studies on bubble pump for a water–lithium bromide vapor absorption refrigeration, International Journal of Refrigeration 21 (6) 452–462, 1998.

[4] A.D. Delano, Analysis of the Einstein Refrigeration Cycle, M.Sc Thesis, Georgia Institute of Technology, 1997.

[5] A.D. Delano, Design Analysis of the Einstein Refrigeration Cycle, PhD Dissertation, Georgia Institute of Technology, 1998.

[6] A. Koyfman, M. Jelinek, A. Levy, I. Borde, A study on bubble pump performance for diffusion absorption refrigeration system with organic working fluids, in: Proceedings of the Second International Heat Powered Cycles Conference, Paris, 2001.

[7] A. Koyfman, M. Jelinek, A. Levy, I. Borde, Experimental study on bubble pump performance with organic working fluids, in: Proceedings of the Third International Conference on Transport phenomena in Multiphase Systems, Keilce, Poland, 2002.

[8] ASHREA Equipment Handbook, Absorption Air-Conditioning and Refrigeration Equipment, 14, 1983.

[9] I., Horuz, "A Comparison between ammonia-water and water-lithium bromide solutions in vapor absorption refrigeration systems", Int. Comm. Heat Mass Transfer, 25, 711-721, 1998.

[10] S. Rohi, M. Naghash Zadegan, H. Monsef, Experimental Analysis of Bubble Pump LiBr Solution working Fluids for absorption refrigeration system, ,Proc.Eurosim, Ljubljana , Slovenia, 2007.

[11] Ozisik M. Heat transfer—a basic approach. McGraw-Hill Book Company; 1985.

[12] Howell RH, Sauer JH, Coad JW. Principles of HVAC. ASHRAE, Refrigeration Equipment, Section 18.21, 1998.

[13] Kreith F, Bohn MS. Principles of heat transfer. 5th ed. PWS Publishing Company; 1997.

[14] Nusselt in Ozisik M. Heat transfer—a basic approach. McGraw-Hill Book Company; 1985.

[15] G.A. Florides , S.A. Kalogirou , S.A. Tassou , L.C. Wrobel "Design and construction of a LiBr–water absorption machine" Energy Conversion and Management 44, 2483–2508, 2003.

[16] Jakob,M , G,Hawkings , Elements of Heat Transfers 3rd ed. New York : John wiley & sons

[17] Icksoo Kyung , Keith E. Herold , Yong Tae Kang, "Model for absorption of water vapor into aqueous LiBr flowing over a horizontal smooth tube", International Journal of Refrigeration 30, 591-600, 2007.

[18] Lee RJ, DiGuilio RM, Jeter SM, Teja AS. Properties of lithium bromide–water solutions at high temperatures and concentration. II. Density and viscosity. ASHRAE Trans; 96(Pt. 1),709–28, 1990.

LATE PAPER

RAYLANG: A MODELING LANGUAGE FOR PERFORMANCE EVALUATION OF STOCHASTIC DISCRETE-EVENT SYSTEMS

Ali Khalili
Mohammad Abdollahi Azgomi
Performance and Dependability Engineering Lab.,
School of Computer Engineering, Iran University of Science and Technology
Tehran, Iran
E-mail: al_khalili@comp.iust.ac.ir, azgomi@iust.ac.ir

KEYWORDS
Modeling Language, Performance and Dependability Evaluation, Formal Methods, Discrete-Event Systems, Object-Oriented Modeling.

ABSTRACT

In recent years, formal methods have been used as an important tool for performance evaluation and verification of a wide range of systems. In the view points of engineers and practitioners, however, there are still some major difficulties in using formal methods. In this paper, we introduce a new formal modeling language to fill the gaps between object-oriented programming languages (OOPLs) used by engineers and the formalisms used for evaluation and verification purposes. The syntax and semantics of a new object-oriented modeling language for discrete-event systems called RayLang is represented in the paper. We have designed the syntax of RayLang similar to OOPLs. In RayLang models, objects that are instantiated from classes, run concurrently and can communicate with each other by requesting services. Every object in RayLang models has some internal state variables and has also some service handlers for executing requests of other objects. We have shown that Markovian RayLang models can be transformed into CTMCs and then be solved by existing solution techniques. For modeling, discrete-event simulation and analytic solution of RayLang models, we have implemented these models in the PDETool framework.

INTRODUCTION

In spite of recent successful application of formal methods, there are still clear needs of further research to develop and design new formalisms. In many notations, it is not easy to understand the meaning and properties of the symbols and how they may and may not be manipulated, and to gain fluency in using them, to express new problems, solutions and proofs (Bowen and Hinchey 2006). The major problem is the difference in the abstraction level and the existing gap between the programming languages used by programmers and software engineers and the modeling languages used for performance evaluation and verification purposes. Most of the existing formalisms cannot be easily used by software engineers. On the other hand, languages used by software engineers cannot be used directly for performance and dependability evaluation purpose and are too informal or too heavy to be analyzed by a verification tool (Sirjani et al. 2005).

The natural idea behind the object-oriented paradigm is to consider the system we intend to model, simulate or develop, as a collection of active objects which collaborate with each other. The term active object denotes an autonomous entity equipped with its own behavior and some internal data which corresponds to the state of object and of course may change during its lifetime (Guelfi et al. 1997).

The aim has been to introduce a new formal modeling language to fill the gaps between object-oriented programming languages (OOPLs) used by engineers and the formalisms used for performance evaluation and verification purposes.

This paper presents the syntax and semantics of a new object-oriented modeling language for discrete-event systems called RayLang. Based on the above aim, we have designed the syntax of RayLang similar to OOPLs. In RayLang models, objects that are instantiated from classes, run concurrently and can communicate with each other by requesting services. RayLang has a well-defined formal semantics and incorporates several ingredients of programming languages and lightweight notations. Every object in a model has some internal state variables and some service handlers for executing requests of other objects. The state space of RayLang models can be generated. We have shown that Markovian RayLang models can be transformed into continuous-time markov chains (CTMCs) and then be solved by existing solution techniques.

The remainder of this paper is organized as follows. Section 2 reports the related works. In section 3, the core language features of RayLang and its syntax and semantics are presented. Analysis of RayLang models are discussed in section 4. Some examples are brought in section 5. Section 6 compares RayLang with other mod-

els. And finally, section 7 concludes the paper.

RELATED WORKS

In the literature, there has been a lot of effort to propose modeling languages for modeling concurrent and distributed systems. It this section, we will briefly review those models, which are closely related to the work presented in this paper.

Petri nets (PNs) (Peterson 1981) which have been introduced for modeling concurrent and distributed systems are well-known models that have graphical representation in addition to the formal definitions and structures. Due to their modeling power and simplicity, Petri nets have been extended with several features. Stochastic extensions of Petri nets, such as stochastic Petri nets (SPNs) (Molloy 1982), generalized stochastic Petri nets (GSPNs) (Ajmone Marsan et al. 1986) and stochastic activity networks (SANs) (Movaghar 1985) are the most famous models used in performance and dependability evaluation of systems.

In addition, several high level and object-oriented extensions of Petri nets have been introduced to make them more appropriate and easy to use. Coloured Petri nets (CPNs) (Jensen 1992), stochastic well-formed nets (SWNs) (Chiola et al. 1993) and language for object-oriented Petri nets (LOOPN) (Lakos and Keen 1991), object Petri nets (OPNs) (Lakos 1995) and concurrent object-oriented Petri nets (CO-OPN) (Buchs and Guelfi 1991) are the most useful examples of such extensions.

Modeling, specification and evaluation language (MOSEL) (Begain et al. 2001) is a description language which depicts a network like model with a controlled flow of tokens, expresses the two-dimensional network structure, the token flow-control mechanisms as well as the routing probabilities and the stochastic delay information in a one-dimensional textual notation.

Maude (Lincoln et al. 1994) is a language based on rewriting logic with modules use rewrite theories, while computation with such modules corresponds to efficient deduction by rewriting. Real-time Maude (Csaba ölveczky and Meseguer 2000) is a language and tool supporting the formal specification and analysis of real-time and hybrid systems in timed modules and timed object-oriented modules which can be transformed into equivalent Maude modules.

Classical process algebras are abstract languages used for the specification and design of concurrent systems. The most widely known process algebras are calculus of communicating systems (CCS) (Milner 1982) and communicating sequential processes (CSP) (Hoare 1978). There exist several timed and stochastic extensions of process algebra with the aim of performance evaluation. Performance evaluation process algebra (PEPA) (Hillston 1996) extends the classical process algebra with the capacity of assigning rates to activities to quantify time and uncertainty, which are described in an abstract model of a system.

The specification language χ (Hofkamp and Rooda 2007) which is developed as a modeling and simulation tool for design of concurrent systems is inspired by CSP and the guarded command language (GCL). Similar to CSP, the behavior of system components is described by processes that communicate via channels. Communication in χ is synchronous, unidirectional and timeless and variables are typed.

The modeling language MoDeST (Bohnenkamp et al. 2003) reasons about stochastic timed systems based on formalism stochastic timed automata (STA) by combining features of probabilistic GCL (pGCL), process algebras with TCSP-like synchronization over common actions, and other language constructs such as urgent actions, exception handling and process instantiation.

The Actor model of computation was originally proposed by Hewitt (Hewitt 1972) and further developed by Agha (Agha et al. 1997) into a concurrent object-based model. Actors are self-contained, concurrently interacting entities of a distributed computing system. They communicate via asynchronous message passing which is fair and can be dynamically created and the topology of actor system can change dynamically. Reactive objects language (Rebeca) (Sirjani et al. 2004) is an actor-based language with a formal foundation supported by a front-end tool for the translation of models into existing model-checking languages.

DEFINITIONS OF RAYLANG

RayLang is an object-oriented stochastic modeling language for discrete-event systems, i.e. systems which are in a state during some time interval, after which an atomic event might happen that changes the state of the system immediately. RayLang has two different settings:

- *Non-deterministic setting.* In this setting, the events occur in a nondeterministic manner and no timing parameter is required. The application of this setting is on model checking.

- *Stochastic setting.* In this setting, the time parameters are specified for the events, which indicate the completion time of the actions related to the events (which are called services here). The application of this setting is on performance and dependability evaluation.

The focus in this paper is only on the stochastic setting. For this purpose, we will firstly present the informal definition of RayLang. Then, the syntax and formal definitions and behavior of the model will be presented.

An Informal Description of RayLang

In RayLang, a model is composed of a finite set of reactive, self-contained and communicating *objects*, which are executed concurrently. An object is instantiated from a *class*, consisting of some *state variables* and *services*. A state variable has a *type* (such as, *integer, short, Boolean,* etc.) which defines the values that variable can hold and should have an *initial value*. A special kind of variables are called *condition variables* that are of type Boolean. *References* are another kind of variables which denote object acquaintances and can be used for object communication purpose.

Services model the behavior of an object change its state variables. Unlike methods in programming language, services have not any return value. There are two types of services: *ordinary* and *immediate*. Ordinary services are executed by passing an asynchronous message (called *service request*) to them. Each ordinary service has an unbounded buffer, identified as *service queue*, for arriving service requests. When a request at the head of a queue of an ordinary service is serviced, its service handler is invoked and the request message is deleted from the corresponding queue after a time based on the service probability distribution function. Each ordinary service could have a precondition which is a Boolean expression and acts as a guard for enabling/disabling execution of the service. If the precondition of an ordinary service is evaluated to true, it can be triggered by removing a request message from the top of its corresponding queue and results in an atomic execution of its body which cannot be interleaved by any other ordinary service execution. For an ordinary service, a user-defined *probability distribution function* (PDF) must be specified which models the service execution time (the time that must be elapsed while the service is enabled until it finishes serving a request in its requests queue).

On the other hand, immediate services do not have any service queue, but rather act like methods or functions in programming languages, i.e. when an immediate service is called (requested), it is executed immediately (synchronously). An immediate service could not have a precondition and is always enabled and can be executed immediately whenever it is called. Requesting an immediate service can be regarded as a synchronous (local or remote) method invocation.

In a RayLang model, computation takes place by means of requesting services (a kind of message passing) and execution of the corresponding services. An object can communicate and interact with other objects when it *"knows"* their name, which are said to be their acquaintances. An object knows its acquaintances by having references to them.

```
model            → (class)+ main rewards
class            → 'class' ClassName '{' (varDefinitions)? servDefinitions '}'
varDefinitions   → varType VarName (',' VarName)* ';'
varType  → refVariableType | conditionVariableType | stateVariableType
refVariableType  → ClassName
condVariableType→ 'condition'
stateVariableType → 'int' | 'short' | 'byte' | 'bool'

servDefinitions  → service+
service          →('immediate')? 'service' ServiceName '('(parameters)? ')' (Rate)?
                        '{' (precondition)? serviceBody '}'
parameters       → parameter (',' parameters )?
parameter        → parType parName
precondition     → 'precondition' ':' booleanExpression
Rate             →':' distributionFunction '(' actualParams ')'

serviceBody      → caseStructre | ( statement ';' )*
caseStructute    → 'case' '{' (option)+ '}'
option           → ProbValue ':' ( statement ';')+
statement        → assgmntStatement | condExp | serviceRequest | brdcstRequets | case

brdcstRequets    → className '.' servName '(' (actualParams)? ')'
serviceRequest   → ((objectID | 'this') '.' )? servName '(' (actualParams)? ')'
actualParam      → ( value | varName | 'this') {',' ( value | varName | 'this') }*

assgmntStatement → varName '++' | varName '--' | varName '=' Expression
condExp          → 'if' '(' booleanExpresion ')' '{' statemensts '}'
                        ('else' '{' statemensts '}')?

main             → 'main' '(' ')' '{' objectDef * '}'
objectDef        → className objName '(' initializationParameters ')' ';'
```

Figure 1: The Syntax of RayLang

The Syntax of RayLang

The Java-like syntax of RayLang classes (object templates), objects (class instantiations), and models (parallel composition of objects) in a BNF-like notation is presented in Figure 1. Note that the words in italic show terminals (such as language keywords, operators and identifiers).

In each class, after declaring variables, some services will be defined. Service body instructions allow a probabilistic case (*case*) or some (remote or local, ordinary or immediate) service requests (*servRequest*), broadcast service requests (*brdcstRequest*), value assignments (*assgmntStatement*), conditional statements (*condStatement*) and sequential compositions.

In service request statements, the ID of callee (object which receives the request) and the service name can be specified that are followed by actual parameters. If the callee is not specified (or keyword *this* is used), the caller and callee are the same which models a local service request. If the requested service is an ordinary service, this can be viewed as an asynchronous message passing that message contains the parameters passed to the corresponding service handler in callee which represents a request for service. This could be regarded as a method call (similar to conventional object-oriented programming language) if the requested service is an immediate service. Sometimes, this can be used for object synchronization.

Ordinary services could be exploited in broadcast service requests. A broadcast request statement $className.serviceName(p_1, ..., p_n)$ causes broadcasting a request message to the ordinary service *serviceName* of all objects in the model with type *className*

where $p_1, ..., p_n$ are parameters of the message. In this way, a request will be sent to an ordinary service of all objects with a special type defined in the model.
A (probabilistic) case statement (*case*) can be used for execution of some alternatives and causes one of its corresponding choices to execute probabilistically. Case statements can only be used in ordinary services. After defining the classes, there is a part for specifying model configuration (*main*). This part is specified with keyword *main* followed by the definition of the model configuration which is defined as a finite number of objects that must be created and then run concurrently. Each class definition has a constructor with a name same as the class. When an object is defined in main, a request is sent to its constructor to create an object with proper parameters. This constructor must assign the initial values for all local (state, condition and reference) variables and could only request local ordinary services (immediate, remote and broadcast service requests as well as case statements are not allowed in constructor). The Constructor must be defined as immediate services. After definition of all objects in main part of a model and thus immediate execution of their constructors, the model will be in its initial state.
Some non-terminals (such as expressions and parameters of services) that are not given in the grammar are like as in programming languages (e.g. Java). Like programming languages, the model uses scope rules for using identifiers. An identifier (of a variable) declared in a class is *"known"* in all services of that class. When an identifier in service parameters has the same name as an identifier in the class, the identifier in the class is *"hidden"* and could be accessed only by keyword this. Because the state and reference variables in an object are private, they could not be used or modified in other objects. Condition variables of an object could be used as read-only Boolean variables via the name of its object (by using its reference) in precondition or statements in service body of other objects. For ease of modeling, the modeler can also use local variables in the body of services which do not affect the object's local configurations (state variables, reference variables etc.). As in programming languages, local variables are created dynamically and are used in execution of a service and after that, the value of them will be destroyed and cannot be used in the next service execution.

Semantics

In this subsection, the semantics of RayLang is presented. The variables defined in an object o_i are typed (integer, boolean, char, condition, reference, etc.). We skip the detail of variable declarations as they are irrelevant to the purposes of this paper. We assume a finite set *Variables* of variables and a domain (type) *Dom(x)* for any variable *x*. We write *Values* to denote the set of all possible values for the variables, i.e. $Values = \bigcup_{x \in Variables} Dom(x)$. The initial value of each variable must be specified in the constructor.
An object o_i (with the unique identifier i) is a tuple $< V_i, Serv_i^*, iServ_i^*, Q_i >$ where V_i is the set of variables (state variables, condition variables and reference variables), $Serv_i^* = \bigcup_{1 \le j \le J} Serv_i^j$ is the set of all ordinary services $Serv_i^j$, and $iServ_i^* = \bigcup_{1 \le k \le K} iServ_i^k$ is the set of all immediate services $iServ_i^k$ of object o_i where J and K are the number of ordinary and immediate services, respectively and j and k are called service identifier of the ordinary service $Serv_i^j$ and immediate service $iServ_i^k$. Each ordinary service $Serv_i^j$ is a tuple $< Pre, Dist, Body >$ where the $Serv_i^j.Pre$ is the precondition expression of the service $Serv_i^j$ (as described in syntax section), $Dist(.)$ is general probability distribution function which may have zero or more parameters and describes the time that the service needs to be executed while it is enabled, and *Body* is a set of RayLang instructions. For an ordinary service, if the precondition is not specified explicitly, it will always be evaluated as true. Each ordinary service in an object has a queue which can be defined as a finite sequence of requests for that service. The service request queue of an object o_i, defined by Q_i, is like a multi-queue consisting of all the queues of its ordinary services and including all the requests that have been sent to the ordinary services and have not been serviced yet. Q^* denotes the set of unbounded FIFO multi-queues that contains message requests of all (ordinary) services of all defined objects, i.e. $Q^* = \bigcup_{\forall i \in I} Q_i$ which Q_i is the unbounded FIFO multi-queue corresponding to services request queue of object o_i. Let's Q_i be a set of all unbounded FIFO queues ($Q_i = \bigcup_{1 \le j \le J} q_i^j = \{q_i^1, ..., q_i^J\}$) where each queue q_i^j contains requests of the ordinary service $Serv_i^j$ in object o_i. For mapping each ordinary service of an object to one of these queues, function *queue*, $queue : Serv_i^j \to q_i^j$, is defined such that $q_i^j \in Q_i$.For a model Ω, there exists a universal set I of all objects that are engaged in the model and $|I|$ denotes the number of all defined objects in the system.
State space of the model (Γ) is defined as:

$$\Gamma = \prod_{1 \le i \le |I|} (\Gamma_i \times Q_i) \tag{1}$$

where $\Gamma_i : Variables \to Values$ is the local state of the object o_i, an evaluation function that maps each local variable to a value of the appropriate type ($\forall x \in Variables, \Gamma_i(x) \in Dom(x)$), and Q_i is the unbounded FIFO multi-queue of services requests for object o_i as mentioned before. However, all of these states do not need to be actually reachable.
Now, we can define the behavior of a model Ω as a labeled transition system $< \Gamma, L, \to, \gamma_0 >$ where:

- $\Gamma = \prod_{1 \le i \le |I|} (\Gamma_i \times Q_i)$ is the set of states as said earlier where Γ_i is the set of possible values for in-

ternal object's (state, condition and reference) variables and Q_i is an unbounded FIFO multi-queue of the service request in object o_i,

- $L = \bigcup_{1 \le i \le |I|} Serv_i^*$ is the set of labels that are all possible ordinary service request that can be executed in Ω, where $Serv_i^j \in L$ means the execution of a request for service $Serv_i^j$ in object o_i,
- $\rightarrow \subseteq \Gamma \times L \times \Gamma$ is the set of transition relations on Γ where $\gamma_1 \xrightarrow{l} \gamma_2$ iff $\gamma_1, \gamma_2 \in \Gamma$ and $l = Serv_i^j \in L$ is an enabled transition which means $\exists j : r = head(\gamma_1.q_i^j)$, r is the request for the ordinary service $Serv_i^j$ on top of the queue and the precondition $Serv_i^j.Pre$ of the service is evaluated as true and γ_2 results from γ_1 by l as follows:
  * Performing the actual transition from state γ_1 to γ_2 (by the service $Serv_i^j$) needs time t which depends on the probability distribution function of service ($Serv_i^j.Dist$) in state γ_1 and the real execution will be completed when time t is elapsed in which the service remains enabled. As we assume that the weight and the priority of all services are equal, in time t, if there exist multiple enabled services that are scheduled for execution, one of them is chosen for execution with equal probability.
  * The request message is removed from the head of $\gamma_1.q_i^j$, i.e. $\gamma_2.q_i^j = tail(\gamma_1.q_i^j)$. This causes assigning the values of the formal parameters to the real parameters, too.
  * An ordinary service execution that is caused by request r leads the execution of service $Serv_i^j$ of the object o_i as a sequence of instructions as an atomic operation as follows:
    - If the ordinary service is a multi-option service (i.e. the service has a probabilistic case statement in its top level of statements), each case option (internal statements of the option) leads in an atomic operation with probability p defined as the option's probability. If not, execution of the service leads to unique outcome state,
    - Execution of an ordinary statement (e.g. assignment statements, conditional statements, etc.) in $Serv_i^j$ may change the value of some variables in object o_i,
    - Execution of each statement $((this|objectID).)?servName(actualParams?)$ (where corresponding *servName* is an ordinary service), changes the service request queue $q_{i\prime}^{j\prime}$ where $i\prime$ is the object identifier of *objectID* and is same as i when local service request (without specifying any *objectID* or with keyword *this*) is used and $q_{i\prime}^{j\prime} \in Q_{i\prime}$ is the queue corresponding to ordinary service *servName* when $j\prime$ is the service identifier,
    - Execution of each ordinary service broadcast request statement (*className.servName(actualParams?)*), changes all service request queues $q_{i\prime}^{j\prime}$ where $i\prime$ is object identifier of all objects with type *className*, $i\prime \in \{t | 1 \le t \le |I| \wedge Dom(o_t) = className\}$, and for each $i\prime$, $j\prime$ will be the service identifier of *servName* in object $o_{i\prime}$. Immediate services could not be used in broadcast requests,
    - Execution of each immediate service request statement $((this|objectID).)servName(actualParams?)$ (where the corresponding *servName* is an immediate service), causes execution of $iServ_{i\prime}^{j\prime}$ (like a method call in programming language) where $i\prime$ is the object identifier of *objectID* and is same as i when local service request (without specifying any *objectID* or with keyword *this*) is used and this causes atomic execution of some statements, respectively.
- γ_0 is the initial state of the model. Variables must be initialized to their default values according to their types in constructors, and Q_0 is defined such that in the beginning of the model, for all i, constructor of object o_i is executed. It is obvious that $\gamma_0 \in \Gamma$.

ANALYSIS OF RAYLANG MODELS

RayLang is a language for modeling discrete-event systems. The state space of RayLang models can be generated by model execution in which every ordinary service can be viewed as an event and the value of object variables in addition to content of service queues represent the state of the model (as mentioned earlier). The state space of the model can be considered as a labeled transition system which could be used in the analysis process. The initial state of the model is the situation where only the constructors of all objects have been executed. In each state, some ordinary services may be enabled, i.e. there exist at least one request in their request queue and their preconditions evaluated to true. Execution of each enabled ordinary service causes a transition to another state (if the modeler uses probabilistic case statement in a service, the execution of the service results in probabilistic transitions into different states). Generating the state space can be done in a depth-first or breadth-first order. Using the defined semantics, if the state space of the model starting from the initial state is finite, we can define the reachability graph as a labeled directed graph based on the reachability set (the set of all reachable states). This reachability graph can be used for model checking. If all ordinary services of a model have exponential PDFs, the reachability graph can also be transformed into a continues-time Markov-chain (CTMC), which can be used for analytic solution

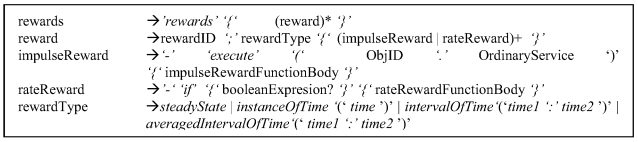


Figure 2: Syntax of Reward Specification in RayLang Models

of the model.
Analytical approaches usually suffer from the state space explosion problem. When the state space of a model is very large or larger than to be handled, discrete-event simulation techniques can be used to analyze RayLang models. A simulation path is a (random) path between some states starting from the initial state. Each event in the path is the (atomic) execution of an enabled ordinary service.

Reward Specification in RayLang Models

For evaluation of a RayLang model, the modeler must define some reward variables which are specific measures of system behavior based on his/her interests. Rewards that are supported here are based on (Meyer and Sanders 1991) which defines a unified approach to the specification of performance, dependability and performability. A reward variable is a set of one or more functions (with type *double*) which return reward value and are defined on a state or transition between states in the model, i.e. there are two types of reward functions: impulse reward functions and rate reward functions that specify impulse and rate rewards respectively. Rate rewards are used to define rewards based on the time in each state and the corresponding function defines the measurement that the reward should evaluate. The impulse reward functions that define impulse rewards are evaluated when the specified ordinary service of an object in the model is executed. For each reward, type of rewards must be defined which determines when, in system time, the reward functions must be evaluated and includes *steadyState*, *instanceOfTime*, *intervalOfTime* and *averagedIntervalOfTime* as presented in the syntax of reward specification (Figure 2).

Implementation of RayLang in PDETool

PDETool is a multi-formalism modeling tool which provides features for construction and translation of models into the XML-based input language of an SDES-based simulation engine, called SimGine (Khalili et al. 2009). PDETool is easily extensible to support a wide range of graphical and non-graphical formalisms. Furthermore, it facilitates construction, animation and analysis of models. A formalism can be implemented in PDETool if a mapping can be provided to the input language of SimGine.

We have implemented RayLang in PDETool by developing a translator which maps RayLang models into the input language of SimGine. It makes modelers enable to evaluate RayLang models using discrete-event simulation technique. If all events' delay functions are exponentially distributed, the tool can also be used to analytically solve the model. In this direction, the model is mapped into a CTMC by generating reduced reachability graph which can be used in both transient and steady-state analysis based on the specified reward variables.
Translation of RayLang model into the input language of SimGine is performed by a two passes *LL(K)* parser. In a nutshell, this process is as follows. Objects' states and services are flattened in the first step of translation. For each object, internal state-variables will be translated into the model's state variable. Each ordinary service is mapped into an event in addition to a queue (as a state-variable) which contains the service's requests. All immediate services are translated into auxiliary function which can be called during model execution and has a special parameter *objectID* to determine the object in which the service is called.

SOME EXAMPLES

In this section, we give two examples to show the syntax and semantics of RayLang.
Example 1. A client-server model. Consider a client-server system, presented in Figure 4, which has a server that serves some clients. Each client sends a request to the server in the service *sendReq* (with exponential rate 1) and waits for the server to respond it. Server has a queue (corresponding to the service *srvReq*) contains all received requests that is not served yet. Process of the request and generating a proper response take a time which is distributed exponentially. The generated response sent to the client which generates the corresponding request and after taht the client *rcvRsp* is executed after 3.0 time units.
There are two classes, client and server, which are templates of the buffer and some clients. Each client has a reference *s* to the object server for interacting with it. Server interacts with its clients via a reference which is passed to the service *rcvReq* by the parameter *c* with of the type of client.
Objects' definition is followed by classes. Each object has a type and proper initialization parameters, which are passed to the constructor of that object. Here, a server and four clients are defined. After initializing all objects by executing their constructors, the model would be in its initial state. The impulse reward variable *reward1* is specified to investigate how many requests are served between time 10 and 20 in the server.
Example 2. Readers and writers. Here, we modify the last example to model probabilistic cases and mutual exclusion. Assume that there exist a buffer and

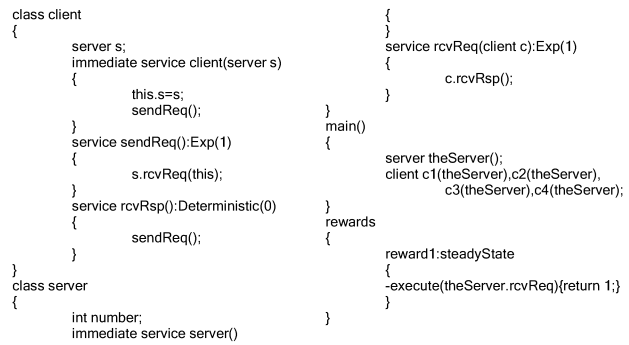


Figure 3: A Client-Server Model in RayLang

some customers that access to the buffer for reading from/writing to it. Multiple readers can read from the buffer simultaneously but a writer could write into it only if there is no readers or writers (any customers generally). Figure 4 shows a RayLang model for this problem.

There are two classes, buffer and customer, which are templates of the buffer and some customers. The state variables *readers* and *writer* in object buffer represent the number of simultaneous readers and writer in the buffer (*writer* is always equal to zero or one). Condition variables *customerExist* and *writerExist* will be true if any customers (readers or writer) and any writer exist in the buffer, respectively.

At the beginning, each object customer will have a request for executing *accessBuffer* to access the buffer. This will cause to try reading from/writing to the buffer with probability 0.9/0.1 which is done by using a probabilistic case statement. Generating such a request is performed during the time exponentially distributed with rate 1. The object *buffer* serves the requests of customers. A service *startRead* can be executed if the buffer is not performing a write request and a service *startWrite* can be started only if the buffer is not performing any other (read/write) request. Executing a service *startRead*/*startWrite* and *finishRead*/*finishWrite* show that the buffer starts to perform a read/write request and finishes performing a read/write request, respectively. Start process of the request takes place as early as possible (PDF of the service is deterministic with parameter zero and it will be done whenever it is enabled). Immediate service *getResult* in each customer object is performed when the request of the object in the buffer is served and change the state of customer from waiting into active.

In reward specification part, *avgNR* is defined as a rate reward which is intended to evaluate the steady state average number of requests exist for reading in request queue of service *startRead* in object *b*. It uses operator # which means the number of requests in the service queue. This property can also be evaluated by reward *avgNR2* which examine the property using variable *readers* in object *b* that explicitly models (counts) the number of read requests. The average number of the write operation, the average number of requests waiting for a read operation and write operation can be evaluated by rewards *avgNW*, *avgNWR* and *avgNWW*, respectively. The simulation results with 99% of confidence level within the confidence interval 0.1 compared to the analytic results (both obtained using PDETool) are presented in Table 1.



Figure 4: A Readers-Writers Model in RayLang

Table 1: Evaluation of Rewards in Readers-Writers Model

Reward ID	Simulation result	Variance	Analytical result
avgNR	2.5715	0.0274	2.5344
avgNW	0.0978	0.0067	0.0998
avgNR2	2.5715	0.0274	2.5344
avgNW2	0.0978	0.0067	0.0998
avgNRW	0.1622	0.0133	0.0164
avgNWW	0.6618	0.0158	0.7060

COMPARISON

The main motivation of designing RayLang was to provide an object-oriented language for modeling, performance evaluation and verification of stochastic discrete-event systems with a well-defined and easy to use syntax and semantics. RayLang has the basic necessary constructs in a programming language-like syntax (instead of using mathematical notation) which is easy to use for practitioners. Thus, it provides an effective and simple approach to model discrete-event systems. It also integrates model based performance evaluation/formal verification techniques and software engineering concepts. To model a discrete-event system using RayLang, a modeler can exploit performance evaluation techniques

based on the specified rewards using analytic solving approaches or discrete-event simulation. Having these features, we can think RayLang as a light-weight notation with a rigid semantics that tries to fill the gap between traditional formal methods and software engineering tools.
In contrast to most modeling formalisms such as Petri nets and process algebras in which the modelers deal with some low-level primitives (e.g. transitions, places, tokens and channels etc.), in RayLang, the important primitives such as objects, variables and services are high-level. This makes RayLang appropriate for modeling systems with a proper abstraction level.
In the next subsection, we will compare RayLang with Rebeca.

RayLang vs. Rebeca

Rebeca (Sirjani et al. 2004), as mentioned in section 2, is an actor-based language for modeling concurrent and distributed systems. If we ignore stochastic (timing) concept of RayLang, it has some conceptual and syntactical similarity with Rebeca. Even, one may think RayLang as an (stochastic) extension of Rebeca with deep modifications. In RayLang, we have *objects*, *services* and *references* instead of *rebecs*, *massage servers* and *known objects* in Rebeca, respectively. The major differences between these two languages can be explained as follows:

- In Rebeca, each rebec has its own message queue and its internal message servers and all requests for the rebec come into this queue. For the request on the top of the queue, the rebec executes the proper message server. But in RayLang as an object-oriented modeling language, each service can be considered as an active thread which has its own request queue that contains the requests sent for the corresponding service.

- RayLang has two types of services: ordinary services and immediate services. In contrast of Rebeca in which all communications take place by asynchronous message passing and a message server may be executed when a request exists on top of the rebec's queue, in RayLang, ordinary services could be executed if there exists a request in its request queue and its precondition is enabled. Immediate services act like (synchronous remote or local) method calls in programming language and execute immediately whenever they are called.

- RayLang supports additional useful modeling facilities (like broadcast service requests and dynamic objects relationship) which may help modeling realistic systems like service-oriented systems and computer networks.

CONCLUSIONS

In this paper, we introduced an object-oriented modeling language called RayLang for stochastic discrete-event systems and formally defined its syntax and semantics. Object-oriented modeling can help modelers through encapsulated constructs. On the other hand, model-based performance and dependability evaluation and formal verification can be used to design more dependable systems. In RayLang, there are active objects (which are instantiated from classes) run concurrently and communicate with each other by requesting services. Each object has its internal variables and thus its internal states. After defining classes and objects, the modeler can evaluate interested measures specified by some reward variables.
The stochastic setting of RayLang models, which we discussed in this paper, can be used for performance and dependability evaluation of stochastic discrete-event systems. The non-deterministic setting of the language can be used for modeling and verification of discrete-event systems via model checking techniques of the model against some specified properties.
We have implemented RayLang within the PDETool by developing a translator which maps RayLang models into the input language of the tool's simulation engine, called SimGine. Numerical analysis of RayLang models (where the model satisfies Markovian properties) can be exploited by generating state space of the model in the tool and solving the resulted CTMC.
Currently, we allow simple data types. More object-oriented properties (e.g. inheritance) and supporting rich data types can be added to the language in future extensions of RayLang.

REFERENCES

Agha G.; Mason I.; Smith S.; and Talcott C., 1997. *A foundation for actor computation. Journal of Functional Programming*, 7, no. 01, 1–72.

Ajmone Marsan M.; Balbo G.; and Conte G., 1986. *Performance models of multiprocessor systems. MIT Press Series in Computer Systems The MIT Press*, 6, 142–188.

Begain K.; Bolch G.; and Herold H., 2001. *Practical performance modeling: application of the MOSEL language*. Kluwer Academic Pub.

Bohnenkamp H.; Hermanns H.; Katoen J.; and Klaren R., 2003. *The modest modeling tool and its implementation. Lecture Notes in Computer Science*, 116–133.

Bowen J. and Hinchey M., 2006. *Ten commandments of formal methods... ten years later. Computer journal of IEEE Computer Society*, 39, no. 1, 40–48.

Buchs D. and Guelfi N., 1991. *A concurrent object-oriented Petri nets approach for system specification.* In *12th International Conference on Application and Theory of Petri Nets.* 432–454.

Chiola G.; Dutheillet C.; Franceschinis G.; and Haddad S., 1993. *Stochastic well-formed colored nets and symmetric modelingapplications. IEEE Transactions on Computers*, 42, no. 11, 1343–1360.

Csaba ölveczky P. and Meseguer J., 2000. *Real-Time Maude: A tool for simulating and analyzing real-time and hybrid systems. Electronic Notes in Theoretical Computer Science*, 36, 361–382.

Guelfi N.; Biberstein O.; Buchs D.; Canver E.; Gaudel M.; von Henke F.; and Schwier D., 1997. *Comparison of object-oriented formal methods.* Tech. rep., Second Year Report of Esprit Long Term Research Project 20072.

Hewitt C., 1972. *Description and theoretical analysis (using schemata) of PLANNER: A language for proving theorems and manipulating models in a robot.* Tech. rep., Artificial Intelligence Technical Report 258, Department of Computer Science, MIT.

Hillston J., 1996. *A compositional approach to performance modelling.* Cambridge University Press.

Hoare C., 1978. *Communicating sequential processes. Communications of the ACM*, 21, 666–677.

Hofkamp A. and Rooda J., 2007. *Chi 1.0 reference manual.* Systems Engineering Group, Eindhoven University of Technology, rev-1322 edition.

Jensen K., 1992. *Coloured Petri NetsBasic Concepts, Analysis Methods and Practical Use. Volume 1-3. Monographs in Theoretical Computer Science An EATCS Series Springer-Verlag*, 97, 141–144.

Khalili A.; Bidgoly A.; and Azgomi M., 2009. *PDETool: A Multi-formalism Modeling Tool for Discrete-Event Systems Based on SDES Description.* In *30th International Conference on Application and Theory of Petri Nets and Other Models of Concurrency, PETRI NETS 2009.* Springer, Paris, France, 343–352.

Lakos C., 1995. *From coloured Petri nets to object Petri nets.* In *16th International Conference on Application and theory of Petri nets, Turin, Italy, June 26-30, 1995: proceedings.* Springer-Verlag, 278–297.

Lakos C. and Keen C., 1991. *LOOPN-Language for object-oriented Petri nets.* In *SCS Multi conference on Object-Oriented Simulation, Simulation Series 23.* Anaheim, California, 22–30.

Lincoln P.; Martí-Oliet N.; and Meseguer J., 1994. *Specification, transformation, and programming of concurrent systems in rewriting logic.* In *Specification of Parallel Algorithms, DIMACS Workshop.* 309–339.

Meyer J. and Sanders W., 1991. *A unified approach for specifying measures of performance, dependability, and performability.* In *Dependable Computing for Critical Applications.* Springer Verlag, vol. 4, 215–237.

Milner R., 1982. *A calculus of communicating systems.* Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Molloy M., 1982. *Performance analysis using stochastic Petri nets. IEEE Transactions on Computers*, 100, no. 31, 913–917.

Movaghar A., 1985. *Performability modeling with stochastic activity networks.* Ph.D. thesis, University of Michigan Ann Arbor, MI, USA.

Peterson J., 1981. *Petri net theory and the modeling of systems.* Prentice Hall PTR Upper Saddle River, NJ, USA.

Sirjani M.; de Boer F.; Movaghar A.; and Shali A., 2005. *Extended Rebeca: a component-based actor language with synchronous message passing.* In *Application of Concurrency to System Design, 2005. ACSD 2005. Fifth International Conference on.* 212–221.

Sirjani M.; Movaghar A.; Shali A.; and de Boer F., 2004. *Modeling and verification of reactive systems using Rebeca. Fundamenta Informaticae*, 63, no. 4, 385–410.

BIOGRAPHY

Ali Khalili received the B.Sc. degree in Computer engineering (software) from Ferdowsi University of Mashhad (2006) and M.Sc. degree in Software Engineering from Iran University of Science and Technology (Sep. 2009). Title of his M.Sc. thesis was "An Object-Oriented Modeling Language for Stochastic Discrete-Event System," which has been done under supervision of Dr. Mohammad Abdollahi Azgomi.

Mohammad Abdollahi Azgomi received the B.Sc., M.Sc. and Ph.D. degrees in computer engineering (software) (1991, 1996 and 2005, respectively) from Sharif University of Technology, Tehran, Iran. His research interests include performance and dependability modelling with high-level modelling formalisms such as stochastic Petri nets, tools for modelling and evaluation, verification and validation, object-oriented modelling, web services, network and web security. He has published several papers in international journals and conferences. Dr. Abdollahi Azgomi is currently an assistant professor at the school of computer engineering, Iran University of Science and Technology, Tehran, Iran.

AUTHOR LISTING

AUTHOR LISTING

Adiguna J. **35**
Aghajani Z. **60**
Al-Akaidi M. **72**
Al-Azzawi L.K.M. **49**
Alshaban S. **49**
Avram D. **105**
Azgomi M.A. **5/60/67/119**
Azziz U. **35**

Béchennec J.-L. **18**
Bidgoly A.J. **5**
Boissière J. **91**
Briday M. **18**
Bruniaux P. **105**

Cabanillas E. **84**

Ehsandar A. **11**
El Falou W. **84**
El-Chahal G. **79**

Farshad Manesh M. **110**

Habchi G. **91**
Hamzaoui R. **72**

Karakchi R.E. **40**
Karcher A. **29**
Kassem R. **18**
Khalili A. **5/119**
Khazan G. **67**

Lafon P. **79**
Lehmann T. **29**

Monsef H. **110**

Naghashzadegan M. **110**
Neagu I. **105**

Peronnard P. **84**

Roohi S. **110**
Rozenberg (Neagu) A. . **105**

Sadkhan S. **49**
Sallez Y. **98**
Sarkarfarshi M. **11**
SavatonG. **18**
Shakeel Ahmad S. **72**
Shmeleva T.R. **53**

Tafazzoli-Shadpour M. . **11**
Tounsi J. **91**
Trentesaux D. **98**
Trinquet Y. **18**

Velazco R. **84**

Younes R. **79**

Zaitsev D.A. **53**
Zaitsev I.D. **53**
Zbib N. **98**
Zeki A.M. **35**
Ziade H. **84**